

INTRODUZIONE

In questo corso vedremo come funzionano le tecnologie che hanno permesso lo sviluppo delle applicazioni distribuite, ovvero di quelle applicazione in esecuzione su diversi calcolatori nel mondo che comunicano tra loro.

Le principali domande di alto livello alle quali vogliamo dare una risposta sono:

1. Quali sono le **tecnologie di accesso** a banda larga e come funzionano?
2. Come garantire un determinato livello di **qualità del servizio** in Internet (*che è nata come rete best - effort*)?
3. Quali architetture e tecnologie permettono di offrire **servizi specifici** in Internet?
4. Quali sono le tecnologie emergenti per **operare** e **gestire** le reti moderne?

INTERNET

Quando nel 29 Ottobre 1969 ci fu la nascita del precursore di Internet, ovvero di ArpaNet, il fenomeno non riscosse molto successo e passò molto in sordina. La rete venne progettata per aiutare l'atterraggio sulla luna americano.

In seguito, negli anni 70, vennero implementate alcune migliorie alla rete e venne implementata AlohaNet, una rete che riprendeva i principi di ArpaNet ma funzionava via radio piuttosto che via cavo. Nel 72 nasce anche il primo programma di posta elettronica e il primo vero protocollo Internet, ovvero NCP, il precursore di TCP/IP (*livello di trasporto*).

Vennero poi definiti anche i principi dell'**internetworking**, ovvero dei principi atti a far comunicare delle reti di reti e nacque **Ethernet**.

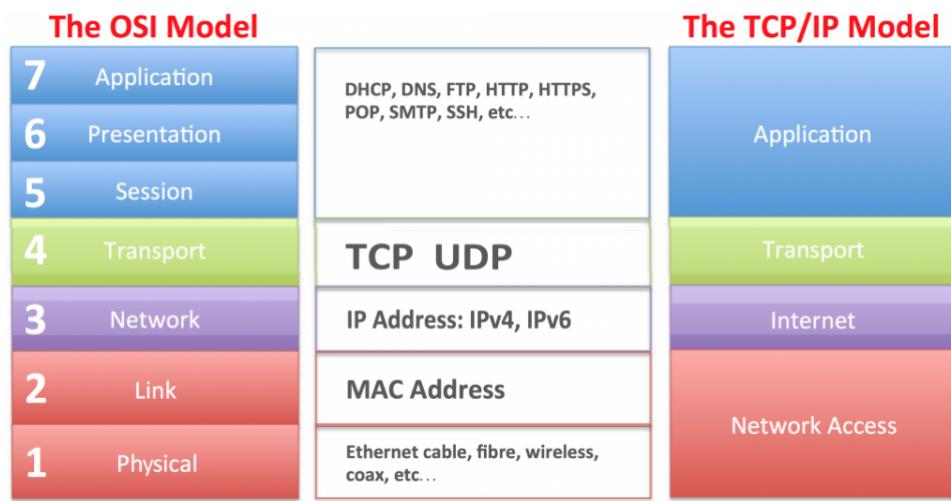
Negli anni 80 nacquero ulteriori protocolli come SMTP, TCP/IP che sostituisce NCP, DNS. Negli anni 90 nasce anche il concetto di navigare in internet tramite browser e si inizia a commercializzare l'Web.

Oggi ci si riferisce ad Internet come alla *rete delle reti*, ovvero come ad un insieme di reti interconnesse a commutazione di pacchetti. Ha una struttura gerarchica dove le varie reti sono raggruppate in sistemi autonomi (AS), ognuno con una propria amministrazione e gestione.

Dal punto di vista dei servizi Internet è una **rete logica** indipendente dalle tecniche trasmissive utilizzate e una **piattaforma** a supporto di applicazioni distribuite indipendente dalle tecnologie di rete e dalla loro evoluzione.

PILA INTERNET TCP/IP E MODELLO A STRATI ISO/OSI

Come sappiamo, la comunicazione su rete è organizzata a strati dove ogni livello esegue certe operazioni specifiche. Lo stack TCP/IP prende spunto da quello ISO/OSI ma lo semplifica per le applicazioni. La rete logica è ottenuta interconnettendo reti fisiche in grado di comunicare fino a livello 3 grazie al protocollo IP. Le entità fondamentali che permettono questa interconnessione sono i **router**, nodi che interconnettono reti fisiche instradando verso la destinazione i vari pacchetti IP, e gli **host**, ovvero i nodi terminali in grado di interpretare tutti i livelli della pila ISO/OSI.



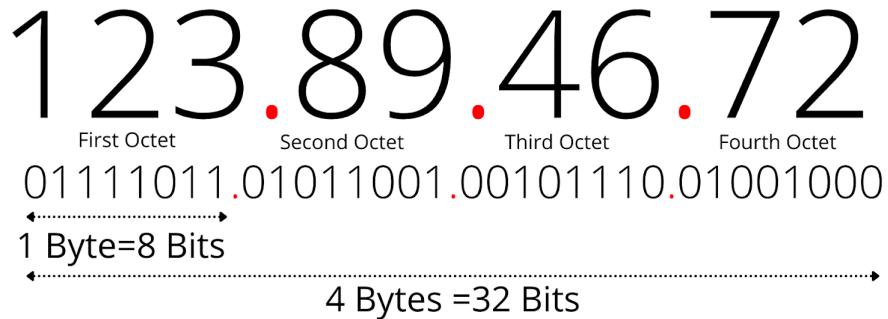
PROTOCOLLI DI RETE: IPv4

I protocolli fondamentali di rete e di trasporto delle reti odierne sono TCP, UDP e IP. Iniziamo col vedere il protocollo di livello 3, ovvero *Internet Protocol*.

Questo protocollo si occupa dell'assegnazione di un indirizzo universale (*indirizzo IP*) ad un'**interfaccia** di rete. Non garantisce integrità, consegna e sequenza dei pacchetti ma effettua una consegna **best effort** dei pacchetti. Esegue la frammentazione dei pacchetti solo se il livello 2 locale lo richiede e ricostruisce i frammenti sono in ricezione.

Come detto, IP è un protocollo di livello 3 (*Rete*) e si appoggia dunque sopra i protocolli di livello 2 (*Data Link*) delle reti che serve che chiaramente possono essere diversi tra i nodi comunicanti (*pensiamo ad esempio ad una rete cablata che comunica con una rete wireless*).

IPv4 Address Format (Dotted Decimal Notation)

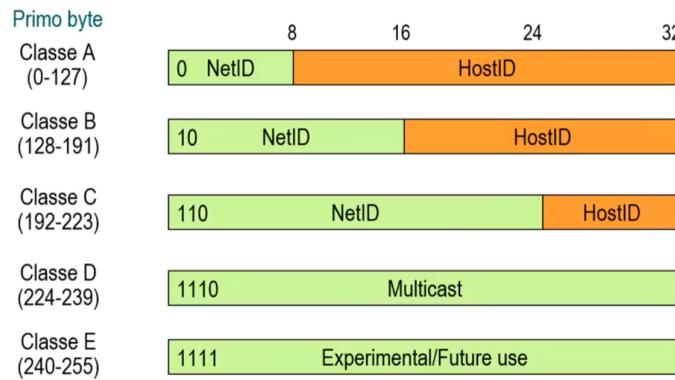


Gli indirizzi IPv4 sono costituiti da 32 bit (*IPv6 ne ha 128*) solitamente raggruppati in gruppi da 8 bit ciascuno (*1 byte*). Per semplicità, sono solitamente rappresentati in notazione decimale puntata dove possono assumere valori da 0 a 255.

Gli indirizzi IP sono logicamente suddivisi in due parti:

- **NetID:** Identifica la rete di appartenenza. Tutti gli indirizzi nella stessa rete hanno lo stesso NetID.
- **HostID:** Identifica l'host specifico.

Alla loro nascita gli indirizzi IP vennero organizzati secondo l'**organizzamento classfull** dove si identificano delle classi basate sui valori dei primi 4 bit dell'indirizzo, andando di fatto a generare una suddivisione maggiore o minore tra la parte di NetID e la parte di HostID. Chiaramente gli indirizzi di classe A erano molto pregiati.



Alcuni indirizzi sono *speciali* tuttavia;

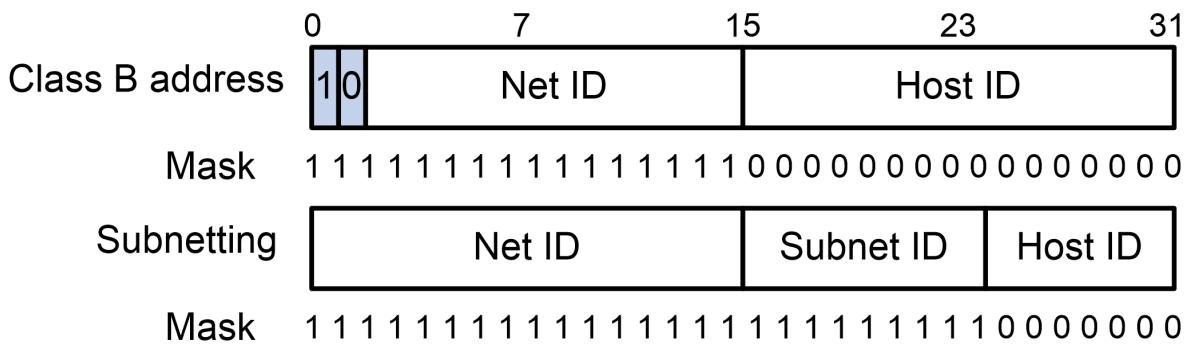
- **Indirizzo di Rete:** L'indirizzo col campo HostID pari a 0 serve ad indicare la rete stessa (*usato nelle tabelle di instradamento*). Non è un indirizzo valido per inviare / ricevere pacchetti.
- **Indirizzo di Broadcast Diretto:** L'indirizzo con il campo HostID posto a tutti 1 indica l'indirizzo di broadcast nella rete indicata dal NetID, ovvero un pacchetto inviato a questo indirizzo verrà ricevuto da tutti i nodi della rete.
- **Indirizzo di Broadcast Limitato:** L'indirizzo con tutti i bit a 1 (255.255.255.255) indica l'indirizzo di broadcast nella rete del mittente del pacchetto. I pacchetti che assumono questi indirizzi non possono oltrepassare i router.
- **Indirizzo di Host:** L'indirizzo col campo NetID posto a 0 indica l'Host che risiede sulla stessa rete del mittente con indirizzo HostID.
- **Loopback:** L'indirizzo con il primo byte pari a 127 indica l'indirizzo di loopback sullo stesso host ed è usato nei sistemi operativi per testare funzionalità di rete.

SUBNETTING

Chiaramente l'indirizzamento classfull non funziona molto bene ai tempi odierni poiché è molto limitante. Per rendere gli indirizzi più flessibili venne quindi inventato il *subnetting*. In particolare, tutti gli host collegati alla stessa rete fisica avranno lo stesso **indirizzo di subnet**.

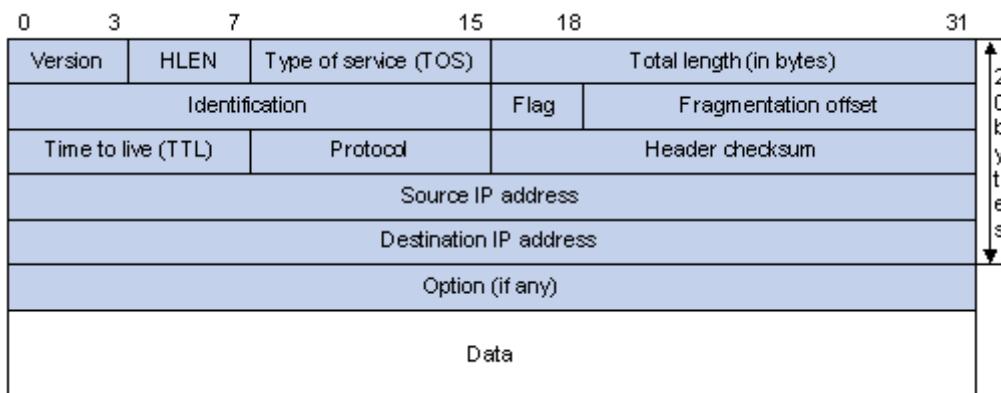
Questo meccanismo, che si basa sulle **subnet mask**, permette una granularità più fine nell'assegnazione degli indirizzi IP.

Una subnet mask permette di suddividere una rete in più sottoreti andando a occupare alcuni bit dedicati all'HostID per identificare la sottorete stessa. In particolare, avendo assegnato n bit dall'HostID al SubnetID, si possono creare 2^n sottoreti ognuna delle quali avrà al massimo $2^{m-n} - 2$ host, dove m è il numero di bit originario dell'HostID. Il subnetting può anche essere applicato ricorsivamente.



PACCHETTO IP

Vediamo ora come è fatto un pacchetto IP concentrandoci su alcuni campi interessanti del suo header.



- Version:** Abbiamo 4 bit che ci indicano la versione di IP usata (*soltanente la 4*)
- HLEN:** 4 bit che indicano la lunghezza dell'header stesso (*comprese opzioni e padding*) espressa in parole da 32 bit. Ha un minimo valore di 5.
- Type Of Service:** Un campo da 8 bit che è usato per la gestione delle priorità nelle code dei router, ovvero vengono usati per implementare la *Quality Of Service*.
- Total Length:** Indica la lunghezza totale del pacchetto (*con tanto di header*).
- Fragment Identification:** Alcuni protocolli di livello inferiore hanno una massima dimensione di trama supportata (*MTU*) inferiore alla massima dimensione di un pacchetto (es. *Ethernet*), dunque, prima di passare il pacchetto a livello inferiore, IP lo divide in frammenti, ciascuno col proprio header. I frammenti saranno poi ricomposti dall'IP del destinatario. I campi *Fragment Identification*, *Flags* e *Fragmentation Offset* sono usati per questo.
- Fragment Offset:** Indica la posizione del frammento nel pacchetto IP originale.
- Flags:** Il primo bit è sempre posto a 0, il secondo **D** sta per *Don't Fragment* (*se risulta necessaria si genera un messaggio d'errore*) e il terzo **M** sta per *More* che è pari a 0 solo per l'ultimo frammento e a 1 per gli altri.
- Time To Live:** Un valore che viene decrementato ad ogni passaggio per un router del pacchetto, in modo che il pacchetto non rimanga in rete indefinitivamente in caso di cattivo instradamento.
- Protocol:** Indica il protocollo di livello superiore (*TCP o UDP*).
- Checksum:** Un checksum dell'header che serve a verificare l'integrità del pacchetto quando viene ricevuto.
- Indirizzi mittente e ricevente.**

PROTOCOLLI RILEVANTI: ADDRESS RESOLUTION PROTOCOL

Per comunicare con un host nella propria rete locale di norma si deve conoscere il suo indirizzo IP ma la comunicazione avviene a livello 2, dove si utilizzano gli indirizzi MAC. Deve quindi essere creata una tabella di corrispondenza tra IP e MAC che viene creata dinamicamente da ciascun host tramite il protocollo ARP.

Per l'acquisizione degli indirizzi MAC nella rete viene inviata una ARP Request in broadcast che include l'indirizzo MAC del mittente in modo che chi la riceve può popolare la propria ARP Cache e l'indirizzo IP dell'host di cui si vuole conoscere il MAC. Successivamente il nodo identificato dall'IP nella Request risponde con un messaggio unicast facendo popolare anche al mittente la propria ARP Cache.

PROTOCOLLI RILEVANTI: DYNAMIC HOST CONFIGURATION PROTOCOL

Può essere comodo non dover configurare i singoli host manualmente con un indirizzo IP, ma sfruttare un server per assegnare gli indirizzi in maniera dinamica. Questo è possibile grazie al fatto che spesso non è necessario avere un'associazione stabile tra indirizzo IP e host ma è possibile avere un'associazione temporanea. Inoltre, spesso gli host sono inattivi e non necessitano di un indirizzo IP. Chiaramente è possibile che all'arrivo di una richiesta non ci siano indirizzi disponibili e in questo caso la richiesta viene rifiutata.

Per questo viene usato il protocollo *client / server* DHCP.

Un client che desidera ottenere un indirizzo IP invia in broadcast un messaggio di *DHCPDISCOVER* contenente il proprio indirizzo fisico (*MAC*) e il server risponde con un *DHCPOFFER* contenente il proprio identificativo e un indirizzo IP proposto.

A questo punto, se il client accetta l'offerta, invia in broadcast una *DHCPREQUEST* contenente l'identificativo del server che creerà e salverà l'associazione tra MAC del client e nuovo IP. Il server infine invia un *DHCPACK* con le informazioni di configurazione necessarie al client (*IP Address, Subnet Mask, Default Gateway, DNS Server*).

Se il client rilascia l'indirizzo invia un messaggio *DHCPRELEASE*, altrimenti l'indirizzo viene rilasciato dopo un timeout.

PROTOCOLLI DI TRASPORTO: UDP E TCP

I protocolli di trasporto sono dei protocolli *end - to - end*, ovvero sono dei protocolli interpretati dagli host e non dai nodi intermedi della rete (*router*).

Questo fa sì che gli host conoscano le applicazioni che richiedono il trasporto di informazioni in rete e conoscono la propria esigenza di capacità di rete. Non conoscono però la capacità di rete effettivamente disponibile e non sono consapevoli della competizione nell'uso delle risorse di rete.

UDP

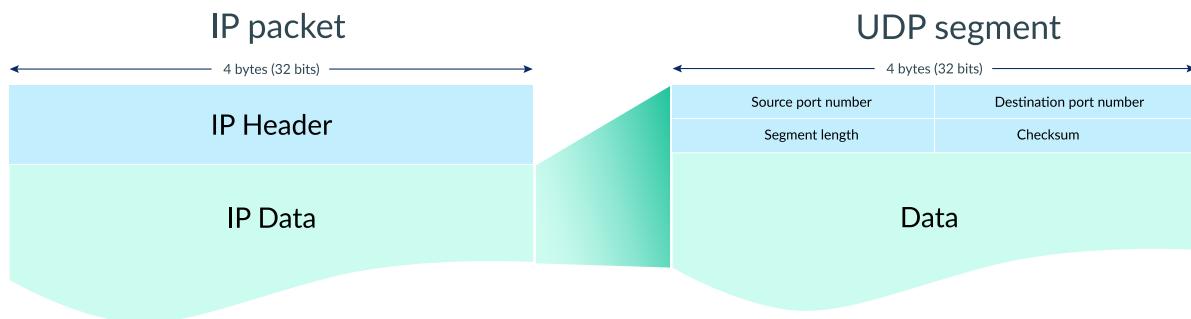
UDP è un protocollo che offre un servizio di tipo **connectionless**, cioè l'invio dei dati in rete non è preceduto da alcuna fase di apertura di una connessione e quindi offre un servizio orientato al messaggio. UDP inoltre è privo di stato, dunque mittente e ricevente non mantengono alcuno stato per scambi di dati e il trasporto offerto in generale è inaffidabile, difatti non offre

rievamento di perdita o duplicazione dei dati, non ritrasmette i dati persi e non fa alcun controllo sulla sequenza dei dati inviata e ricevuta e non esegue alcun controllo di flusso.

Tuttavia UDP offre la **multiplazione** di flussi di pacchetti da / verso più applicazioni contemporaneamente attive in un host (*identifica le applicazione mediante la porta*) e il **rilevamento d'errore** (*su tutto il datagramma*).

UDP in genere è adatto per applicazioni tolleranti alla perdita ma sensibili al ritardo (*es. streaming audio / video*).

Il datagramma UDP è molto minimale: presenta solo le porte di sorgente e destinazione, la lunghezza del datagramma e un opzionale checksum.



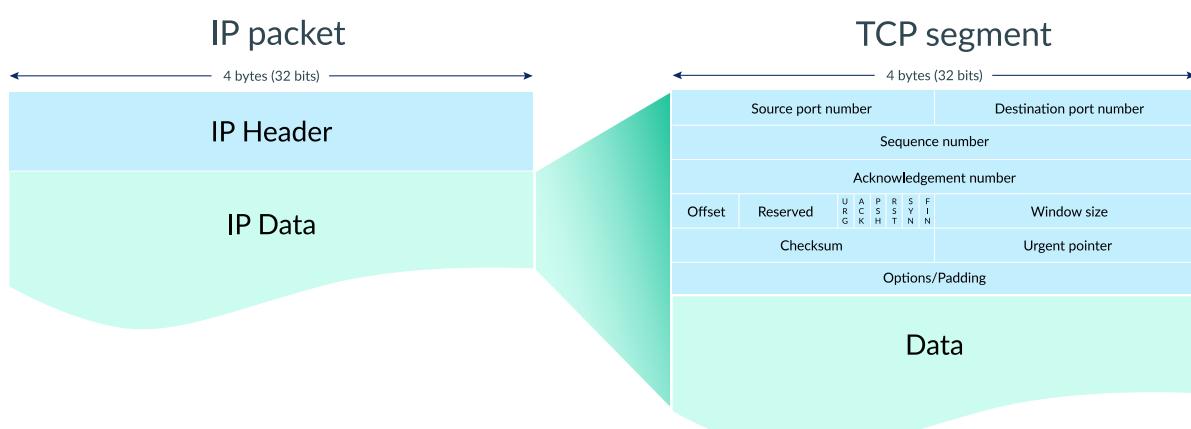
TCP

TCP offre sempre un servizio *end - to - end* ma è orientato alla connessione e quindi richiede una fase iniziale di apertura di un canale di comunicazione (*il messaggio fa parte di un'entità maggiore, il flusso*). Il suo scopo è quindi trasmettere dei dati in maniera *affidabile*.

Per fare ciò offre dei servizi in più rispetto a UDP quali la consegna sequenziale dei pacchetti, il recupero dei dati persi o corrotti, l'eliminazione dei duplicati. Anche TCP offre la multiplazione di più flussi come UDP usando le porte.

Chiaramente, visto che IP è connectionless, i nodi intermedi non sanno nulla della connessione tra sorgente e destinatario.

Essendo TCP più complesso rispetto a UDP, anche il suo datagramma risulta più articolato, in particolare presenta dei *sequence number* utili alla ricostruzione sequenziale dei pacchetti, un *ack number* utilizzato per la verifica e l'eventuale ritrasmissione dei dati (*dice fino a dove ho ricevuto i dati*) e *window size* che indica quanti byte l'applicazione è disposta a ricevere e permette di fare un *controllo di flusso*.



Per l'instaurazione della connessione TCP utilizza un **handshake a tre vie** dove il client invia un messaggio di *SYN* al server che risponde con un suo *SYN + ACK* al quale il client risponde con un *ACK*. Durante questi messaggi viene stabilito anche il *sequence number* di inizio di entrambe le parti (*gli ACK in particolare incrementano il sequence number opposto*).

L'abbattimento della connessione avviene in maniera *graceful*, ovvero con l'utilizzo di un messaggio di *FIN* da parte di entrambe le parti.

Quando si perdono dei messaggi persi o di cui non si è ricevuto *ACK*, TCP ritrasmette tali messaggi. Il controllo d'errore inoltre è obbligatorio in questo caso. Chiaramente questo implica che possono esserci anche dei duplicati (*se a perdersi non è il messaggio stesso ma l'ACK*), ma TCP elimina i duplicati e ripristina la sequenza dei messaggi grazie al *sequence number*.

CONTROLLO DI FLUSSO

Abbiamo detto che un host può specificare quanti dati può ricevere grazie alla *window size*, specificando quindi la **finestra di ricezione**. La quantità di dati che il trasmettitore può inviare (**finestra di trasmissione**) sarà quindi al più pari alla finestra di ricezione del destinatario. Tali finestre possono essere modificate dinamicamente per adattarsi alle capacità di elaborazione degli host.

CONTROLLO DI CONGESTIONE

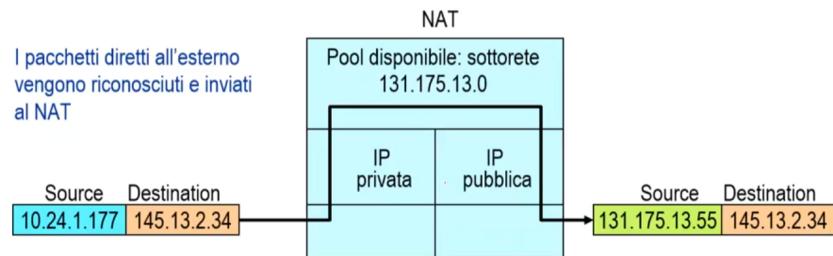
TCP adotta anche una **finestra di congestione** che viene dinamicamente variata in base allo stato di congestione della rete (*stimata sulla base degli ACK non ricevuti*). In questo caso, il mittente invia la quantità di dati pari al **minimo tra finestra di ricezione e finestra di congestione**.

Chiaramente quindi TCP è utilizzato per applicazioni non sensibili al ritardo ma non tolleranti alla perdita di dati (es. *FTP*).

PROTOCOLLI RILEVANTI: NETWORK ADDRESS TRANSLATION

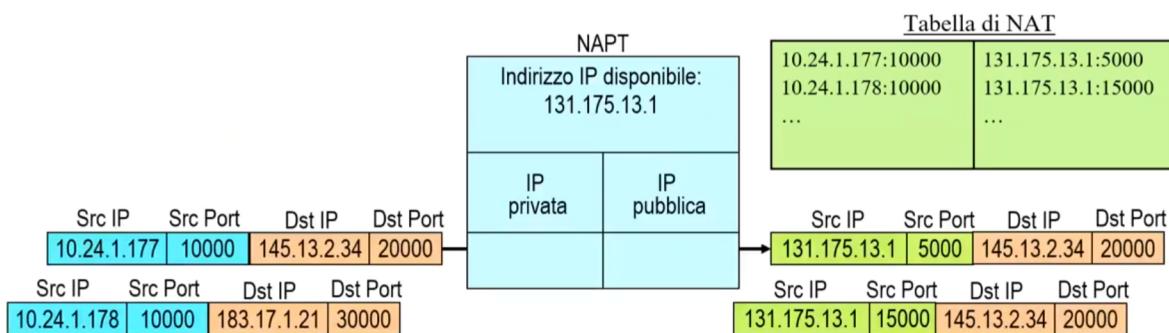
L'aumento esponenziale del numero di host collegati ad Internet ha reso un problema la disponibilità di indirizzi IPv4. Per risolvere questo problema si è trovata una soluzione basata su **indirizzi privati** adottati nelle reti locali private. Chiaramente però i pacchetti con indirizzi privati non possono viaggiare in Internet ed è quindi necessario uno strumento che permetta a host con indirizzi privati di interfacciarsi ad Internet.

Per fare ciò si usa il protocollo NAT che associa un ridotto numero di indirizzi IP pubblici a tanti indirizzi privati associati agli host. Quando un host vuole inviare un pacchetto all'esterno della rete, il NAT intercetta il pacchetto e trasla l'indirizzo mittente con uno di quelli pubblici.



Chiaramente però è necessario che la comunicazione sia bidirezionale e quindi è necessario che il NAT abbia una associazione tra indirizzi privati e pubblici associati. Solitamente si usa una associazione dinamica basata sul meccanismo di **sessione**, dove al termine della sessione (*la connessione TCP viene chiusa*) l'indirizzo pubblico viene rilasciato. Chiaramente però questo può portare ad un blocco a causa dello scarso numero di indirizzi pubblici.

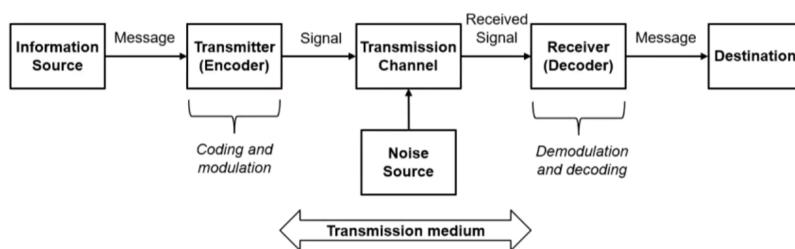
Per questo venne creato anche il **NAPT (Network Address Port Translation)** che trasla la coppia <Indirizzo IP, porta>, permettendo a molti indirizzi interni di usare uno stesso indirizzo pubblico.



UN'INTRODUZIONE ALLA TEORIA DELLA COMUNICAZIONE

Passiamo ora ad un livello più basso per capire come una sequenza di bit possa essere trasmessa su un mezzo comunicativo. Vedremo cos'è un **canale di trasmissione** e come usarlo al meglio tramite tecniche di **codifica** e di **modulazione**, come usarlo per trasmettere dati da molteplici sorgenti tramite il **multiplexing** e le tecniche di **accesso multiplo** e infine quali sono i **mezzi di trasporto fisici** sui quali si può costruire un canale di comunicazione.

Nel 1948 Shannon e Weaver proposero un modello che può essere sempre adottato quando si ha una comunicazione.



In questo modello abbiamo:

- **Sorgente di Informazione**: un'entità che genera informazione e invia dei *messaggi* ad un trasmettitore.
- **Trasmettitore**: sfrutta dei meccanismi di codifica e di modulazione per far sì che il messaggio sia predisposto al meglio per la sua trasmissione sul canale di comunicazione,

trasformandolo in un segnale nel tempo.

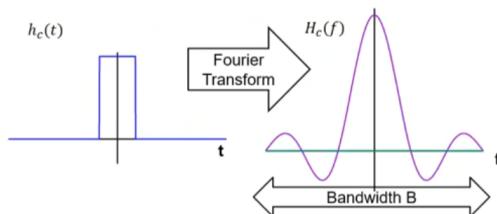
- **Canale di Trasmissione:** convoglia il segnale verso il ricevitore. In questo modello è un'entità astratta, può essere un cavo in rame, in fibra, un trasmettitore radio ma anche una sequenza di canali fisici.
- **Sorgente di Rumore:** agisce in maniera randomica sul canale di trasmissione.
- **Ricevitore:** fa la demodulazione e la decodifica del segnale per ricavarne il messaggio originale.
- **Destinazione:** riceve il messaggio ricostruito.

Ci concentriamo per lo più sulla comunicazione digitale che richiede che i *bit* di informazione siano codificati in **segnali fisici** che possano essere propagati nel mezzo trasmissivo.

Il canale di trasferimento viene modellato tramite una funzione $h_c(t)$ che descrive come questo cambia il segnale d'ingresso $s(t)$ nel segnale d'uscita $\tilde{s}(t)$ tenendo conto del rumore $n(t)$. Inoltre, questa funzione modella anche l'**attenuazione**, ovvero come il mezzo trasmissivo *assorbe* il segnale e la **distorsione**, ovvero la modifica della rappresentazione del segnale.

CAPACITÀ DEL CANALE

Sappiamo che ogni segnale nel tempo può essere osservato come una somma infinita di segnali sinusoidali con una determinata ampiezza e fase. La **trasformata di Fourier**, che prende il segnale nel tempo e lo rappresenta nello spazio delle frequenze, mostra, per ogni componente in frequenza (*armonica, onda sinusoidale*), la sua ampiezza e la sua fase (*rappresentata coi numeri complessi*). Questo ci permette di osservare una **banda di frequenze** che un segnale occupa.



Dunque la **banda di un canale** è il range di frequenze entro il quale i segnali che può trasmettere devono stare.

Data quindi la banda del canale B , si può dimostrare che la massima capacità di un canale C (*massimo bitrate*) tale per cui l'errore è piccolo a piacere può essere calcolata come $C = B \cdot \log_2(1 + S/N)$ dove S ed N sono le potenze del segnale e del rumore che agiscono sul canale.

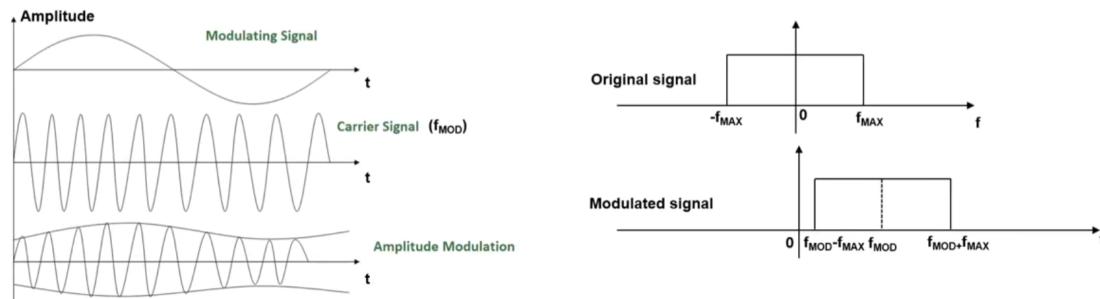
Questo limite è un *upper bound*, l'obiettivo è usare delle buone tecniche di codifica e modulazione che si avvicinino il più possibile a questo limite.

MODULAZIONE

ANALOGICA

Con *modulazione* ci si riferisce a due fenomeni. Il primo di questi è la **modulazione analogica** che consiste nella conversione in **alte frequenze** di un **segnale in banda base**. Questo viene fatto andando a cambiare l'ampiezza, la fase o la frequenza. In questo modo si ottiene un segnale modulato adatto ad essere trasmesso sul mezzo considerato. Questo procedimento viene fatto appunto per occupare certe bande di ampiezza prestabilite coi segnali (*ad esempio la radio che ha varie frequenze per vari canali*).

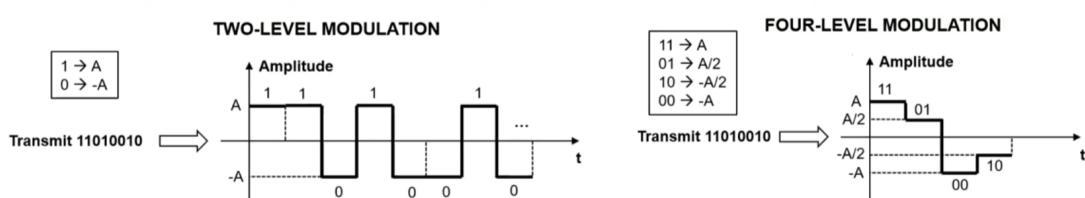
Nell'esempio abbiamo il *segnale modulante* che è il segnale che vogliamo trasmettere e il *segnale portante* con la sua frequenza f_{MOD} . Con la modulazione di ampiezza andiamo a variare l'ampiezza della portante sulla base del segnale da trasmettere.



DIGITALE

La modulazione digitale è sotto certi versi simile a quella analogica. Essa permette di assegnare delle forme d'onda a **gruppi di bit** che devono essere trasmessi con l'obiettivo di ottenere un segnale robusto alla distorsione e al rumore. Ogni forma d'onda è chiamata **simbolo** e dura per un certo periodo di tempo; il numero di simboli trasmessi al secondo è chiamato **baud rate**. Esistono vari schemi caratterizzati dal numero di bit che possono essere trasmessi per simbolo. In generale schemi con molti bit per simbolo (4 - 12) portano ad un alto bit rate ma richiedono un rapporto segnale / rumore alto.

Soltanamente per decidere cosa rappresenta un simbolo si usa un sistema a soglia. Osserviamo inoltre come per la modulazione a quattro livelli non è stata scelta un'organizzazione *intelligente* dei valori assegnati ai simboli: se sbagliamo da $A/2$ a $-A/2$ sbagliamo due bit. Se avessimo assegnato i bit come 11 01 00 10, per ogni possibile sbaglio avremmo sbagliato ad interpretare un unico bit.



CODIFICA

La codifica consiste nel mappare un gruppo di bit ad un altro gruppo di bit per predisporre il segnale al meglio per la sua trasmissione.

Esistono due tipi di codifica:

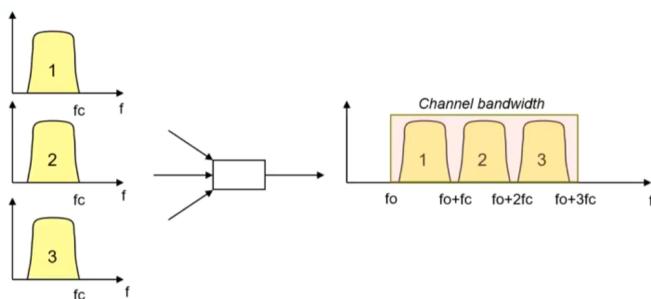
1. **Codifica di Sorgente:** comprime l'informazione per ridurre la ridondanza del segnale originale. Tende a diminuire il numero di bit.
2. **Codifica di Canale:** protegge l'informazione dai *bit error* introdotti dal canale di trasmissione usando tecniche di **error correction** e **detection codes**. Tende ad aumentare il numero di bit.

MULTIPLAZIONE

La multiplazione descrive il concetto di divisione della capacità offerta da un canale di trasmissione in cui quindi più segnali vengono combinati in un unico segnale (es. *il digitale terrestre*). La capacità può essere condivisa andando a dividere ed ad assegnare a differenti sorgenti (*utenti o servizi*) una specifica risorsa, sia essa **frequenza, tempo o codice**. Si hanno chiaramente diverse tecniche di multiplazione classificate in base alla risorsa divisa. Spesso la multiplazione è gerarchica.

Abbiamo vari tipi di multiplazione:

- **Frequency Division (O)FDM** usata spesso anche in contesti analogici. Include anche la *Wavelength Division Multiplexing* utilizzata nelle trasmissioni ottiche.
In questo tipo di multiplazione ogni segnale viene convertito dalla sua banda base ad una specifica banda portante. La demultiplazione avviene usando dei filtri passabanda che vadano a prendere solo le bande relativi ai vari segnali che poi possono essere riportati nella loro banda base.



Una variante, usata spesso nelle reti wireless, è la *Orthogonal FDM* che divide il segnale da trasmettere in un grande numero (es. 1024) di *sottoportanti* diverse che garantiscono un basso bitrate. Queste portanti sono **ortogonalini**, cioè non generano mutua interferenza tra loro.

In questo modo un numero di sottoportanti (*anche non contigue*) può essere assegnato a diverse sorgenti. La trasmissione sulle varie portanti avviene in parallelo.

Il fatto di poter assegnare sottoportanti non contigue permette di far sì che ogni sorgente abbia una qualità decente (*in un mezzo come l'wireless dove la qualità cambia spesso*) poiché gli si può assegnare alcune sottoportanti di buona qualità e alcune di cattiva qualità.

Inoltre, per le portanti di qualità maggiore si può usare una modulazione più aggressiva avendo un bitrate più alto.

Dunque OFDM aggiunge uno strato di flessibilità.

- **Time Division**

- *Synchronous (TDM)* usata di solito in reti a commutazione di circuito.

In questo caso la sorgente può usare tutta la banda disponibile ma per un periodo limitato di tempo.

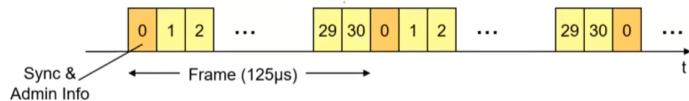
In particolare, nel caso sincrono, si hanno dei **time slot** assegnati a ciascuna sorgente durante i quali questa può trasmettere.

Questi time slot sono organizzati in una struttura periodica che prende il nome di **trama**.

Chiaramente bisogna avere una sincronizzazione tra le sorgenti per capire quando inizia una trama, per cui si ha di solito uno slot di sincronizzazione.

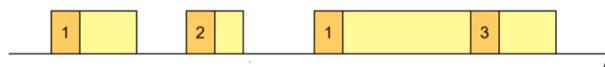
In questo caso è possibile ma molto difficile multiplcare sorgenti con bitrates diversi, per cui di solito si usa per reti a commutazione di circuito.

Un caso tipico che usa questo tipo di multiplazione è la rete telefonica.



- Asynchronous che può essere **slotted** (ATM) o **unslotted** (es. IP).

In questo caso l'informazione è suddivisa in pacchetti che possono avere dimensioni diverse e quindi il tempo non è più slottato ma dipende dal pacchetto (*a meno che i pacchetti abbiano dimensione fissa*). Anche in questo caso si usa tutta la banda.

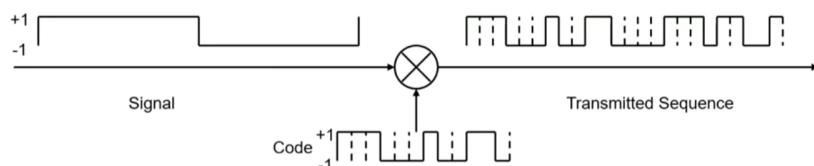


La multiplazione a tempo asincrona abilita la **multiplazione statistica**, ovvero una tecnica che permette di adattare la condivisione del canale in base all'intensità di traffico istantaneo generata da diversi sorgenti. Permette quindi di usare il canale maggiormente a chi ne ha più bisogno in un certo istante di tempo. Chiaramente si rischia di non essere in grado di trasmettere l'informazione di tutti a volte.

Ad esempio se abbiamo un canale a 100 Mbit/s e delle sorgenti con un average rate di 1 Mbit/s e un peak rate di 2 Mbit/s si può pensare di permettere più di 50 sorgenti (*che garantisce che tutti possano inviare dati al loro peak rate*) poiché solitamente non tutte le sorgenti raggiungeranno insieme il peak rate.

• Code Division (CDM)

In questo caso la risorsa condivisa è un *codice*. Anche questo tipo di multiplazione è molto adatta per le trasmissioni radio (*usata in 3G*).



Quello che viene fatto è moltiplicare il segnale per il codice ottenendo delle sequenze diverse da trasmettere. Chiaramente c'è bisogno che i codici assegnati alle diverse sorgenti siano *ortogonal**i*. Tutte le sequenze trasmesse vengono sommate nel mezzo e il ricevitore, andando a moltiplicare il segnale ricevuto per il codice usato dal mittente, riesce a ricostruire solo il messaggio di interesse.

ACCESSO MULTIPLO

L'accesso multiplo è la condivisione di una capacità offerta da un canale trasmittivo in modo concorrente tra sorgenti diverse.

A differenza della multiplazione, le sorgenti sono collegate allo stesso mezzo di comunicazione e ci vogliono trasmettere in modo concorrente, come ad esempio le radio che si contendono l'etere.

Anche per l'accesso multiplo abbiamo diverse tipologie:

- **Tecniche di Canalizzazione**, molto simili alle loro controparti di multiplexing

- Frequency Division (FDMA)

Le portanti (*sotto - portanti*) sono dinamicamente assegnate alle sorgenti.

- Time Division (TDMA)

I time slot sono assegnati dinamicamente alle sorgenti.

- Code Division (*CDMA*)

I segnali, moltiplicati per diversi codici, sono trasmessi da diverse sorgenti.

- **Tecniche Random**

- Aloha e Slotted Aloha

In Aloha quando una sorgente deve trasmettere, trasmette. Se c'è una collisione ritrasmette dopo un tempo casuale.

Nella versione slottata non si può trasmettere in un qualsiasi istante di tempo, ma solo in determinati slot.

- CSMA (*CSMA, CSMA/CD, CSMA/CA*)

Migliorano le tecniche Aloha in quanto prima di trasmettere si effettua un ascolto del canale per identificare e/o cercare di evitare le collisioni.

- **Tecniche Controllate**

- Reservation

Chi vuole trasmettere chiede di poterlo fare.

Nel momento in cui può farlo sa di essere autorizzato e quindi sa di non creare collisioni.

- Polling

Caso in cui un dispositivo *chiede* ad un altro dispositivo se ha qualcosa da trasmettere o no. Usato per esempio in Bluetooth.

MEZZI TRASMISSIVI

In genere la trasmissione si divide in due categorie: **trasmissione guidata** che permette la propagazione di un segnale in un mezzo fisico (*cavi coassiali, twisted pair, fibre ottiche*) e **trasmissione radio** dove il segnale è propagato nell'aria dove possibilmente incontra vari ostacoli.

TWISTED PAIR (*DOPPINO IN RAME*)

Un doppino è composto da una coppia di fili di rame isolati che sono intrecciati tra loro in modo da ridurre le interferenze generate da doppini diversi vicini. Se i cavi fossero in parallelo agirebbero come un antenna e sarebbero cioè in grado di trasmettere e ricevere segnali.

Intrecciando i cavi si crea una **interferenza distruttiva** benefica alla comunicazione.

Poiché solitamente in un cavo abbiamo diverse coppie di doppini questi sono intrecciati con degli intervalli di intrecciatura diversi per limitare ancor di più l'interferenza. Si possono poi anche intrecciare le varie coppie tra loro.

I doppini sono generalmente organizzati in **categorie**, ognuna delle quali ha dei requisiti di qualità specifici e che quindi permettono certe velocità di trasferimento. Tuttavia, i doppini usati nella rete telefonica non rientrano in quelle categorie e, per questo, spesso le reti di accesso ne soffrono.

FIBRE OTTICHE

Una fibra ottica è una fibra molto fine fatta di silicio (*vetro*) che può propagare luce vicina agli infrarossi con una **attenuazione molto bassa** usando un metodo di propagazione basato sulla **riflessione interna totale** che usa il nucleo interno molto sottile e il *cladding* che avvolge il nucleo ed è fatto di vetro con un indice rifrattivo minore; dunque quando un segnale raggiunge il cladding viene totalmente riflesso nel core.

La banda disponibile nelle fibre ottiche è quasi illimitata (*maggiori di 100 GHz*) e per questo vengono spesso utilizzate nelle dorsali.

Esistono due tipi di fibre:

- **Multimodali:**

Hanno un nucleo di spesso circa $50\mu m$ e sono *meno pregiate* ma che richiedono dei dispositivi per la trasmissione più economici. Permettono di trasmettere il segnale con molteplici modi di propagazione ma che portano ad una dispersione del segnale nel tempo. Tuttavia presentano un limite per quanto riguarda la banda e la distanza.

- **Monomodali:**

Hanno un nucleo di circa $8 - 10\mu m$ e permettono di raggiungere dei rapporti banda - distanza molto più elevati.

ETERE

Abbiamo che l'etere è suddiviso in varie sottobande (*LF, HF, VHF, UHF*) che vengono utilizzate per diverse applicazioni. Quelle che ci interessano maggiormente sono le **UHF** e le **SHF** che vanno da 300 MHz a 30 GHz e sono usate per le reti cellulari e le reti satellitari.

Le principali reti wireless sono:

- **Wireless Local Area Networks** che operano tra i 2.4 e i 5 GHz (**SHF**) e subiscono interferenze da altri sistemi (*microonde, telecomandi, ecc.*). A 5GHz soffrono anche di attenuazione dovuta a pioggia e/o nebbia.
Le bande usate da queste reti inoltre sono **non licenziate**, cioè chiunque può trasmettervi senza nessuna autorizzazione.
- **Mobile Radio Networks (2G, 3G, 4G, 5G)** che operano tra gli 800 MHz e i 2.6 GHz (**UHF**) e che possono ricoprire grandi distanze con poca potenza.
- **Reti Satellitari e Rete 5G Radio Network**: operano tra i 3 e i 30 GHz e usano wave millimetriche per frequenze sopra i 25 GHz. Permettono di avere una grande banda ma subiscono molto l'attenuazione meteorologica.

Chiaramente per comunicare in etere si usano le **antenne** che irradiano energia elettromagnetica nello spazio e la catturano anche. Le antenne possono irradiare energia in maniera uniforme (*radiatore isotropico, anche se sono solo teoriche*) o concentrate in alcune direzioni (*antenne omnidirezionali come quelle dei telefoni o direzionali come quelle dei tralicci*). Esistono anche delle *antenne settoriali* che coprono una certa area sul piano orizzontale (*ad esempio 120 gradi*).

Chiaramente l'etere è **broadcast** per sua natura e quindi qualsiasi segnale inviato può essere captato da qualsiasi ricevitore e quindi spesso si adottano delle architetture centralizzate dove i nodi possono comunicare solo col master.

I segnali nell'etere subiscono vari problemi come l'attenuazione dovuta alla distanza (*presente anche nella trasmissione guidata*) e la **variazione delle caratteristiche del canale nel tempo**.

L'attenuazione viene modellata dalla *free - space attenuation* $P_r = P_t \left(\frac{\lambda}{4\pi d} \right)^2$ dove

P_r è la potenza ricevuta, P_t è la potenza trasmessa e d la distanza tra trasmettitore e ricevitore. Chiaramente l'attenuazione è tanto più grande quanto è grande la distanza ed è tanto minore quanto più grande è la **lunghezza d'onda** (*o minore la frequenza per trasmettere*).

Altri parametri che portano problemi alla trasmissione in etere sono la riflessione del segnale, la diffrazione di questo, lo *shadowing* e lo scattering,



Questi effetti portano a due fenomeni:

1. **Multi - Path Fading:** il segnale può dividersi e percorrere diversi percorsi dal trasmettitore al ricevitore e le repliche si sommano e si possono verificare interferenze costruttive o distruttive, portando ad una qualità del segnale che varia molto nel tempo.
2. **Shadow Fading:** il segnale viene interrotto da oggetti grandi che ne degradano la qualità in istanti di tempo larghi.

Esistono anche delle contromisure per questi problemi come il **power control** che stabilisce che il segnale trasmesso dalle varie stazioni deve avere una potenza controllata per non disturbare le altre comunicazioni. Altre tecniche sono ad esempio la modulazione adattiva.

Esiste quindi una relazione tra frequenza di trasmissione e distanza di attenuazione: in particolare, maggiore è la frequenza, minore è la distanza necessaria ad attenuare il segnale. Dunque, per trasmissioni a frequenze alte sono necessarie più antenne.

Nelle reti wireless inoltre viene usata la multimazione nella direzione **downlink** (*dall'antenna al nodo*), mentre in **uplink** viene usato l'accesso multiplo.

Un concetto simile alla multiplazione è il **duplexing** che consiste nella separazione delle risorse del canale (*tempo e frequenza*) per uplink e downlink.

Nelle reti vengono usate anche le tecniche **MIMO (Multiple Input Multiple Output)** che sfruttano la *multiplazione spaziale* per avere velocità di trasmissione più elevate o una grande resistenza al rumore e al multipath fading. Questo viene fatto trasmettendo lo stesso stream da più antenne.

RETI D'ACCESSO A BANDA LARGA

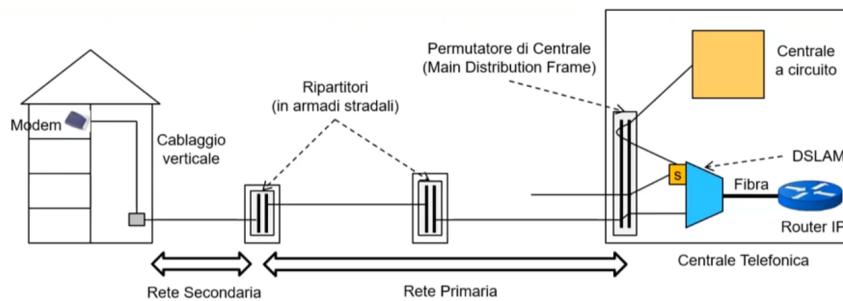
Le **reti d'accesso** sono quelle parti di rete che raccolgono il traffico da parte degli *utenti periferici*. Per *reti a banda larga* si intende reti veloci e quindi con tanta banda disponibile (*chiaramente la definizione non è univoca e dipende dagli standard attuali*).

Abbiamo vari tipi di reti di accesso:

- **Reti di accesso fisse:** cablaggio fino a casa dell'utente (*rame, fibra, fibra misto rame*). La mobilità dell'utente non è prevista.
- **Reti di accesso wireless a postazione fissa:** cablaggio fino ad un *Point Of Presence (POP)* e poi comunicazione wireless. Anche qui non è prevista una grande mobilità dell'utente.
- **Reti di accesso satellitare:** cablaggio fino ad una stazione radio di terra e poi comunicazione veicolata da un satellite verso la casa dell'utente. Mobilità dell'utente non prevista.
- **Reti di accesso radiomobile:** cablaggio (*o ponte radio*) fino alla stazione radio base, poi comunicazione radio fino a terminale mobile. In questo caso è prevista la mobilità

dell'utente.

RETI DI ACCESSO FISSE



La prima rete di accesso che vedremo è quella in rame che sfrutta l'infrastruttura della rete telefonica pre-esistente e quindi sfrutta i doppini di bassa qualità per il trasferimento dati (*rendendola però una soluzione economica*). Notiamo che dal punto di vista strutturale abbiamo, alla postazione dell'utente, un *cablaggio verticale* che arriva ad un *punto di distribuzione* da cui parte una rete secondaria che arriva ad un *ripartitore*. Diversi ripartitori creano il collegamento alla centrale creando la *rete primaria*. Concettualmente quindi esiste un collegamento in rame dalla casa dell'utente fino alla centrale, anche se fisicamente ci sono i ripartitori di mezzo.

Alla centrale c'è un *permutatore di centrale* che rappresenta la terminazione di tutte le linee degli utenti e che le collega ad un **DSLAM (DSL Access Multiplexer)** che multipla le varie linee utente verso una fibra in modo che poi i dati vengano convogliati in internet. Il DSLAM funziona secondo un concetto di modulazione e demodulazione ed è per questo che è richiesto avere un modem anche a casa dell'utente. Inoltre, il DSLAM si occupa anche di splittare il segnale telefonico dal segnale dati.

Chiaramente la distanza tra la centrale e la casa dell'utente impatta sulla qualità della trasmissione, così come anche la qualità dei ripartitori.

xDSL

L'idea di base alle tecnologie xDSL è di sfruttare al massimo la banda disponibile sui doppini telefonici già installati (*che tradizionalmente era filtrata 3.4 KHz*). Esiste un tradeoff tra **banda disponibile** e **lunghezza del doppino**, difatti le tecnologie più *spinte* xDSL vanno bene solo per distanze ridotte.

In particolare, le tecnologie ADSL utilizzano delle bande diverse per upstream e downstream poiché quando è stato definito lo standard era più importante il downstream. Le tecnologie VDSL permettono invece diversi schemi di suddivisione della banda. Chiaramente la banda disponibile parte dai 4 KHz, dove finisce la banda della rete telefonica.

G.fast utilizza un time division duplexing e quindi utilizza tutta la banda solo se tutti i doppini adiacenti usano G.fast e non altre tecnologie xDSL per evitare di creare interferenze.

Tecnologia	Standard	Banda	Capacità (down/up o aggregato)	Distanza massima (indicativa)	
ADSL2+	G.992.5	2,2 MHz	24/1,4 Mbit/s	Qualche km (circa 3/3,5 km)	Possibilità di ripartire la banda in modo più o meno simmetrico
VDSL2	G.993.2 17a	17,6 MHz	70 Mbit/s	Qualche centinaia di m (circa 500m)	
VDSL2	G.993.2 30a	30 MHz	100+ Mbit/s	Qualche centinaia di m (circa 500m)	
VDSL2	G.993.2 35b	35 MHz	200+ Mbit/s	Qualche centinaia di m (circa 500m)	
G.fast	G.9700/9701	106/212 MHz	700 Mbit/s	Qualche decina di m (meno di 100m)	

La tecnica che sta alla base delle xDSL è il **vectoring** che è una tecnologia che elimina la mutua interferenza tra doppini adiacenti e migliora la qualità della comunicazione. Chiaramente quando si trasmette un segnale, questo arriverà al ricevente voluto, ma anche ad altri riceventi sotto forma di interferenza con una qualche attenuazione. Questo si può modellare come $r_1 = s_1 T$

dove $T = \begin{bmatrix} 1 & k_2 \\ k_1 & 1 \end{bmatrix}$ e i k indicano i coefficienti che rappresentano l'interferenza mutua.

L'idea è di trasferire, invece del segnale direttamente, il segnale $s^* = T^{-1}s$ dove T rappresenta la matrice dei coefficienti di attenuazione che modellano cosa ricevono gli altri riceventi, in modo che il ricevente possa ricevere esattamente s . Chiaramente i k sono stimati. I vantaggi sono molto limitati nel caso di più operatori telefonici sulla stessa linea.

FIBRA MISTO RAME

Abbiamo visto che esiste un tradeoff tra lunghezza del doppino e banda disponibile: per risolvere questo problema si usano delle reti in fibra misto rame dove le distanze maggiori vengono coperte con la fibra, lasciando solo le tratte più corte al rame. L'idea è spostare la fibra il più vicino all'utente (**FTTx**).

La rete in rame vista prima in particolare è una **Fiber - to - the - Exchange (FTTE)** dove la rete in fibra arriva solo fino alla centrale telefonica.

Ci sono poi delle reti **Fiber - to - the - Cabinet (FTTC)** dove la rete in fibra arriva fino al cabinet che contiene anche un *Multi - Service Access Node* con un *Mini DSLAM*. I cabinet stanno attivi e quindi devono essere alimentati.

Altre possibilità sono le **Fiber - to - the - Building (FTTB)** dove la rete in fibra arriva fino all'edificio dell'utente, lasciando al rame solo il cablaggio verticale. Il *DSLAM* sarà quindi installato nell'edificio. Infine ci sono le reti **Fiber - to - the - Home (FTTH)** dove la fibra arriva fino all'utente.

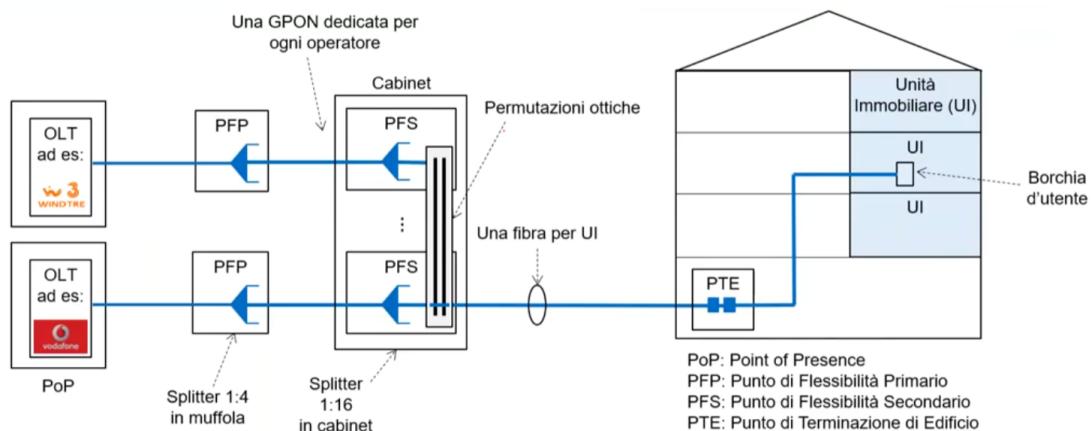
Esistono le reti FTTH **Point - to - Point (P2P)** che però vengono usate solitamente per utenze business poiché richiede molta fibra e forniscono molta banda. La multiplazione / demultiplazione avviene in un concentratore (*MPoP*).

Spesso però per le FTTH e le FTTB vengono usate delle **Passive Optical Networks (PON)** in cui vengono usati degli *splitter* passivi che ripartiscono il segnale in più rami in modo da diminuire il numero di fibre pur condividendo la banda tra gli utenti. Solitamente lo splitting avviene a due livelli. Si usano due dispositivi attivi, un **Optical Network Unit (ONU)** e un **Optical Line Termination (OLT)**.

La trasmissione in downstream avviene in broadcast mentre l'upstream avviene mediante accesso multiplo TDMA governato dall'OLT. Le reti PON possono essere usate anche per le FTTC ma è poco diffuso perché si ha un doppio livello di condivisione delle risorse. Tuttavia, le ONU sono dispositivi attivi e quindi costano, specialmente se sono una per edificio. Per risolvere questo problema si utilizza una tecnica di **Reverse Power Feeding** in cui la ONU viene alimentata direttamente dal doppino in rame dell'utente.

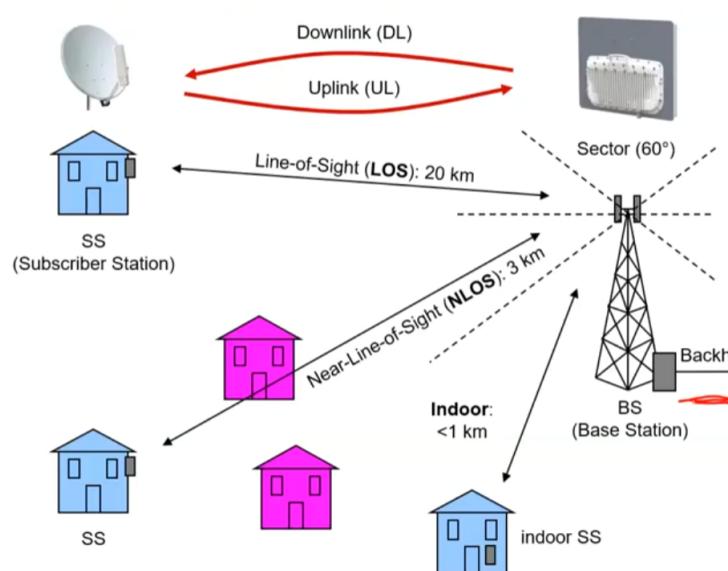
Ci sono soluzioni come **Open Fiber** che è un operatore *wholesale only* che stende la rete in fibra e poi vende i propri servizi di connettività agli operatori di rete. In particolare, nella soluzione Open Fiber, si ha una terminazione di edificio dove c'è una interconnessione uno a uno tra le fibre entranti e quelle uscenti e non si ha quindi splitting a questo livello.

Da questo punto si passa al cabinet dove ci sono gli *splitter*, che prendono il nome di *punto di flessibilità secondario* e che sono uno per operatore (*soltanente lo splitting ratio è di 1:16*). Si ha inoltre nel cabinet un elemento che permette le permutazioni ottiche in modo che si possa favorire il passaggio di un utente da un operatore ad un altro. Si hanno poi dei *punti di flessibilità primari* interrati in delle muffole che portano le fibre alle OLT degli operatori. Questo tipo di architettura offre una grande flessibilità in quanto tutte le operazioni possono essere svolte nei cabinet e non è necessario rientrare negli edifici.



RETI DI ACCESSO WIRELESS A POSTAZIONE FISSA (FWA)

Queste reti sono spesso definite anche come **Fiber - to - the - Tower (FTTT)** o **fibra mista radio** e rappresentano un'alternativa più economica e flessibile rispetto alle reti di accesso fisso. Sono quindi molto diffuse in zone montane e rurali, zone in cui sarebbe anti - economico cablare una nuova rete fissa, ed è quindi adatta a ridurre il **digital divide**.



Si hanno principalmente tre modalità di propagazione:

1. **Line Of Sight (LOS):** trasmissione in linea diretta (*senza ostacoli*), solitamente usata in ambienti aperti. Chiaramente permette buone prestazioni a distanze maggiori.
2. **Near Line Of Sight (NLOS):** ci sono ostacoli tra trasmettitore e ricevitore ma si è sempre in ambiente aperto. Riduce la distanza avendo comunque delle buone prestazioni.
3. **Indoor:** trasmissione in ambiente chiuso. L'antenna stessa è interna e quindi le distanze sono molto ridotte.

Si utilizzano delle **antenne fortemente direzionali** per *LOS* ed *NLOS* e delle **antenne omnidirezionali** per *Indoor*. Per le stazioni base si usano delle **antenne settoriali** con tecnologia MIMO. Operano intorno ai 5 GHz e quindi le condizioni meteorologiche possono influenzare il segnale.

RETI DI ACCESSO SATELLITARI

Anche queste reti, come le FWA, cercano di abbattere il digital divide e sono quindi utilizzate per fornire connessioni a questi posti difficilmente raggiungibili.



Per funzionare, questo tipo di rete ha bisogno dell'installazione, presso il sito dell'utente, di un **Very Small Aperture Terminal (VSAT)**, ovvero di una parabolica che punta verso un satellite che permette anche il **dual feeding**, ovvero permette anche di puntare verso due satelliti diversi.

Presso la stazione terrestre viene invece installato un paraboloide con grande apertura che fornisce l'accesso alla rete fissa.

Chiaramente questo tipo di reti hanno una **latenza non trascurabile** dovuta alla doppia comunicazione tramite satellite. La distanza del satellite influisce sulla connessione e satelliti lontani non sono utilizzabili per applicazioni real time.

Abbiamo vari tipi di satelliti:

- **Geostazionari:** distanti circa 35800 km, buoni per la trasmissione TV ma non adatti ad applicazioni real time. Presentano latenze di centinaia di millisecondi ($> 500\ ms$). Il loro piano orbitale coincide col piano equatoriale della Terra e quindi hanno la stessa velocità della rotazione della Terra, cosa che permette loro di avere una posizione fissa all'orizzonte e quindi di avere una gestione semplice. Non sono in grado di servire i poli.
- **Orbita media:** distanti circa 5000 - 15000 km (*usati ad esempio per GPS*). Non sono in genere usati per Internet.
- **Orbita bassa:** distanti circa 500 - 1000 km, permettono applicazioni real time con una latenza di decine di millisecondi ($< 50\ ms$) ma richiedono una gestione più complessa in quanto i satelliti si muovono in modo molto rapido rispetto alla Terra ed essendo vicini ad essa è necessario avere una copertura ampia.

Le bande utilizzate dalle reti satellitari sono:

- **C:** 4 - 6 GHz, usata anche per la TV
- **Ku:** 11 - 14 GHz
- **Ka:** 20 - 30 GHz

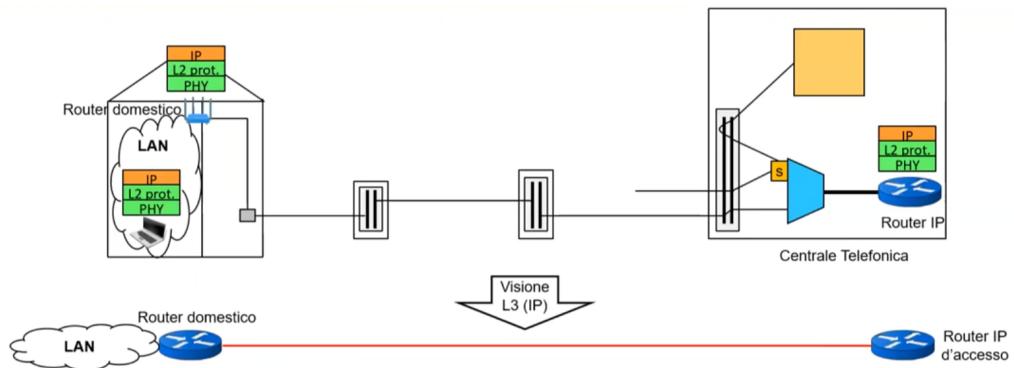
Inoltre i satelliti hanno delle coperture molto ampie (*anche 1/3 della Terra*) usando due tecnologie:

1. **Single Beam:** servono un'area unica con un unico segnale, avendo una banda limitata per singolo utente.
2. **Multibeam:** servono aree diverse con segnali differenti (*spot beam*) gestendo meglio la banda.

Per il downstream viene usata la multiplazione TDM mentre per l'upstream viene usato l'accesso multiplo TDMA o CDMA.

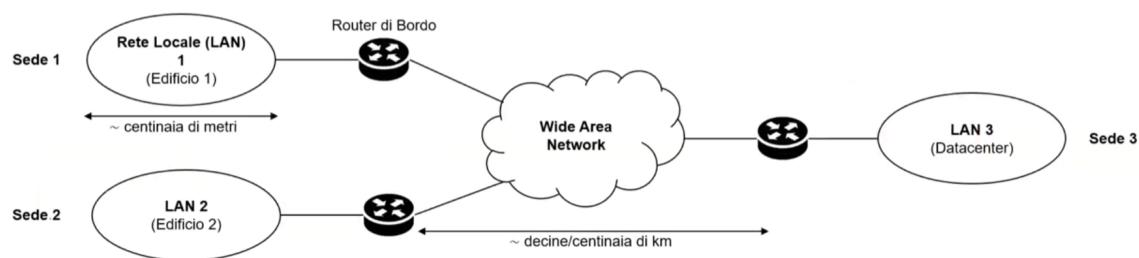
SERVIZI DI CONNETTIVITÀ WAN

Le reti d'accesso viste in precedenza permettono di interconnettere gli utenti ad una **rete IP pubblica** e di accedere a Internet. Se guardiamo le reti d'accesso dal punto di vista dello strato 3 della pila ISO/OSI quello che vediamo è di fatto un *cavo logico* che va dal *router domestico* ad un *router IP di accesso*.



Tipicamente, per gli utenti residenziali, si ha una **connettività generalizzata** dove l'accesso a Internet è fornito da un Internet Service Provider (*ISP*). La rete in questo caso ha una organizzazione gerarchica dove esistono diverse reti organizzate in Sistemi Autonomi (*AS*) che comunicano tra loro mediante dei **router di bordo** e l'instradamento è garantito da protocolli di routing IGP (*intra AS*) ed EGP (*inter AS*).

Per le soluzioni aziendali invece spesso si ha una **connettività dedicata** dove il servizio è pensato per interconnettere sedi di aziende o punti vendita/presenza. Chiaramente questo tipo di soluzione ha dei costi più elevati di svariati ordini di grandezza. Questo tipo di connettività è anche nota come **connettività WAN**. Solitamente le LAN hanno un'estensione di al massimo una centinaia di metri, mentre la WAN si estende per decine/centinaia di km. Anche qui tra le LAN e la WAN ci sono dei router di bordo. Le interconnessioni in caso di connettività dedicata richiedono delle **garanzie** in termini di **qualità del servizio (QoS)** che la rete best - effort Internet non sempre può garantire (es. *latenza end-to-end < 5 ms*).

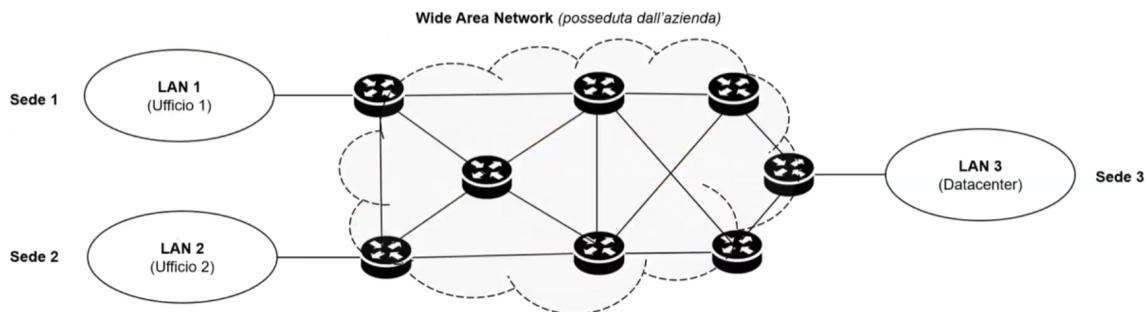


Vedremo ora possibili soluzioni per l'implementazione di una WAN.

WAN FISICA DEDICATA

La soluzione più banale è quella dove l'azienda possiede la Wide Area Network e quindi possiede l'infrastruttura di rete e la gestisce.

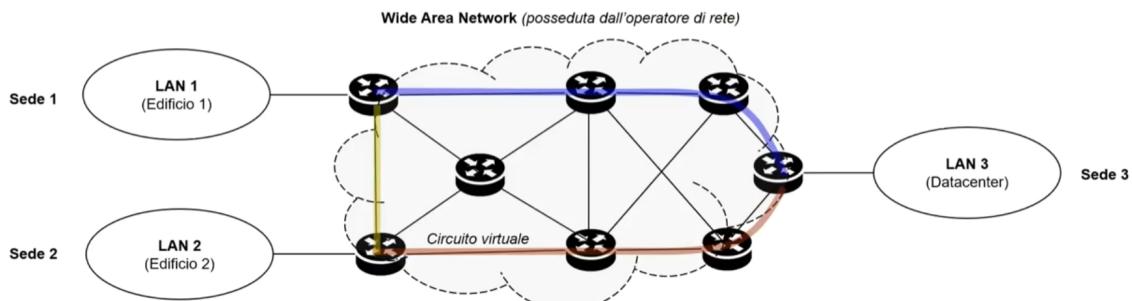
Questo tipo di soluzione permette un controllo completo della rete e una banda molto elevata, ma garantirne la sicurezza è responsabilità dell'azienda. Inoltre, spesso queste soluzioni sono molto costose. A volte si usano le **dark fibers** (*quando un operatore effettua degli scavi spesso posa più fibre del necessario, in modo da affittarle per questo genere di soluzioni*) per ridurre parzialmente i costi.



LINEE DEDICATE

In questo caso l'azienda stipula un contratto con un operatore di rete per l'instaurazione di *circuiti privati* tra le sue sedi. Ad esempio i circuiti dedicati possono riguardare delle **lunghezze d'onda dedicate** sui collegamenti in fibra (*Wavelength Division Multiplexing*).

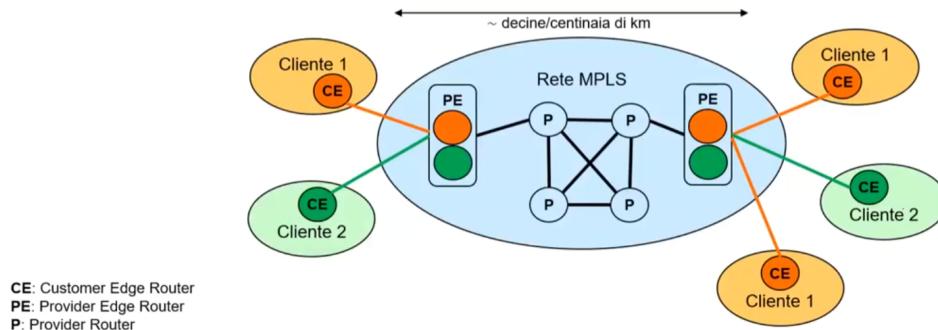
Si ha in questo caso un controllo parziale della rete, una grande banda e la responsabilità per la sicurezza della rete è dell'operatore di rete. Anche questa soluzione può risultare costosa, anche se meno della WAN fisica dedicata.



MULTIPROTOCOL LABEL SWITCHING (MPLS) WAN

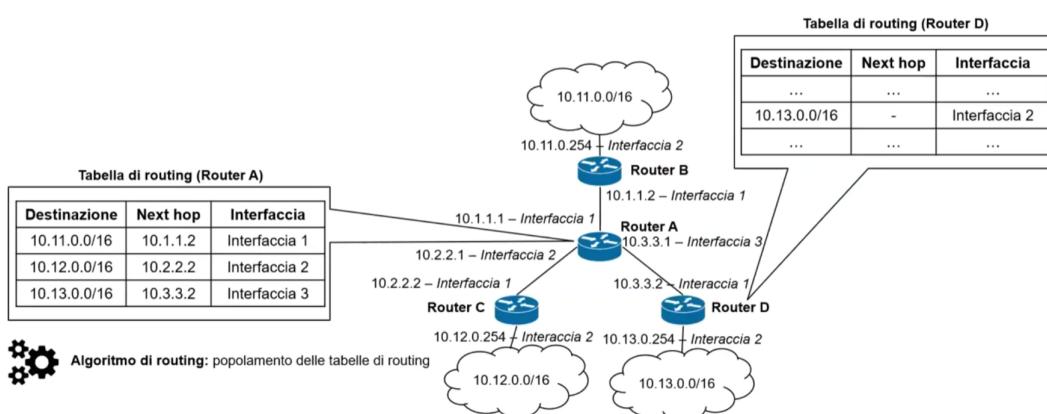
Una soluzione più economica è la *MPLS WAN* dove l'azienda stipula un contratto con un operatore di rete per ottenere una **connettività mesh** con una certa garanzia di QoS. Questa soluzione permette la creazione di una **rete privata virtuale (VPN)**.

I clienti si connettono, mediante un proprio **Customer Edge Router** ad uno stesso **Provider Edge Router**: a questo punto è responsabilità dell'operatore garantire le interconnessioni dei diversi clienti. I costi sono di circa 500\$/Mbps al mese.



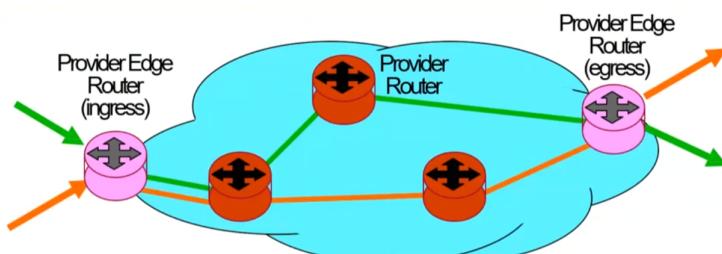
RIPASSO: ROUTING IP

Sappiamo che il routing IP definisce come i pacchetti debbano essere convogliati dalla sorgente alla destinazione. Gli elementi fondamentali utilizzati a tal scopo sono le **tabelle di routing** e gli **algoritmi di routing**. Notiamo che nelle tabelle di routing l'inoltro dei pacchetti avviene sulla base della destinazione: i pacchetti con un determinato prefisso vengono associati ad una determinata interfaccia. Se il next hop è assente significa che il pacchetto può essere direttamente inoltrato dal router.



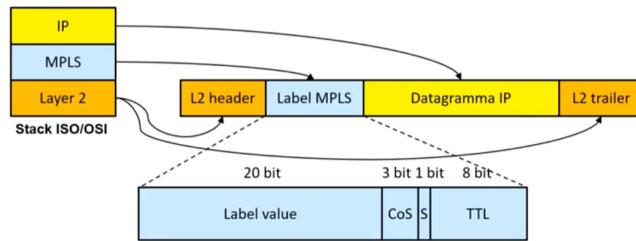
L'architettura MPLS invece permette di definire dei **circuiti virtuali** per i flussi di dati che possono essere **predeterminati** dal gestore o su richiesta esplicita degli utenti o **instaurati all'occorrenza** per mezzo di un meccanismo di set up e di prenotazione delle risorse.

Nelle reti MPLS è possibile ottimizzare dinamicamente l'instradamento dei flussi ed è possibile instradare in base ad un **ricco set di parametri** (sorgenti, porte, applicazioni, ecc) oltre alla destinazione.



LABEL SWAPPING FORWARDING

Si ha quindi un cambio di paradigma rispetto ad IP con il cosiddetto **label swapping forwarding** (*invece del destination based forwarding di IP*) dove il pacchetto IP è incapsulato in un **LS Header**. La **label MPLS** (20 bit) si interpone tra l'header di livello due e l'header IP e può essere annidata a stack.



L'header LS presenta i seguenti campi:

- **Class of Service (CoS)**: permette di prioritizzare certo traffico in caso di congestione.
- **Stack (S)**: consente l'uso in cascata, se vale 1 indica *bottom of stack* e quindi la label più interna.
- **TTL**: simile ad IP. Serve anche qui perché i router MPLS vedono l'header IP come un payload.

Dunque la label è usata per commutare e questo permette la creazione di un **Label Switched Path** (*circuito virtuale*) tra una sorgente e una destinazione (*tra i Provider Edge Routers*).

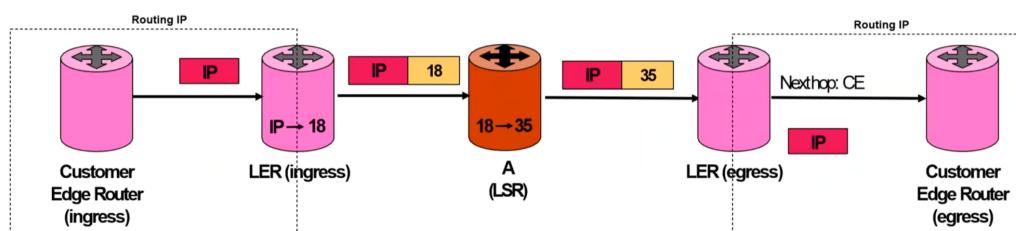
L'associazione tra interfacce e label si trova nella **MPLS Forwarding Table**.

In Interface	In Label	Out Label	Out Interface
...
3	21	18	4
3	56	135	6
...

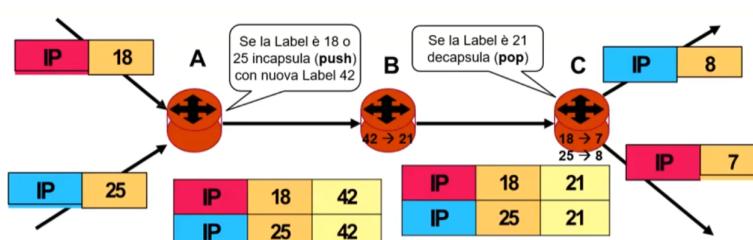
Le reti MPLS nominano i Provider Edge Routers come **Label Edge Routers (LER)** e i Provider Router come **Label Switched Router (LSR)**.

Le label sono modificante dai router nel pacchetto durante la fase di *label swapping* e sono associate tra loro e a determinate interfacce in/out dei differenti router in fase di setup dei cammini.

Notiamo che nella parte di rete tra Customer Edge Router e LER avviene tramite classico instradamento IP, sia in entrata che in uscita. L'incapsulamento delle label avviene nella rete virtuale tra i LER e i LSR.



Nelle reti si può fare, grazie alla possibilità di annidare più header MPLS, l'**affasciamento di cammini**: immaginando di avere due flussi diversi, si possono stackare le label per unire i due flussi all'interno della rete mesh. Il LER A impilerà le label e invierà il flusso, il LER C farà una pop delle label per sapere poi a chi inoltrare i due flussi decapsulati. In questo modo i router interni commutano meno flussi permettendo una maggiore scalabilità.



Si possono creare dei *Label Switched Path* manualmente ma è un'operazione tediosa ed error prone.

Per la creazione automatizzata dei *Label Switched Path* viene disaccoppiato il traffico di controllo da quello dati e questo permette di abilitare il **traffic engineering** che permette i percorsi da intraprendere e far sì che i vari percorsi non interferiscano fra loro.

I pacchetti di controllo vengono utilizzati dunque per la creazione automatizzata dei LSP e seguono un inoltro simile a quello IP.

I router MPLS sfruttano delle componenti assenti nei classici router IP quali:

- **Traffic Engineering Database (TED)**: contiene informazioni topologiche della rete derivate dai protocolli di routing e informazioni sulle risorse di rete (*banda dei link, banda prenotata*) derivate da estensioni dei protocolli di routing. Contiene inoltre anche dei dati amministrativi derivati dalle configurazioni degli utenti. Consente di creare un determinato cammino di rete.
- **Procedure di Segnalazione**: invio e ricezione di messaggi di controllo per la creazione del LSP. Permette di coordinare la distribuzione delle label, instaurare un cammino desiderato, riservare le risorse, riassegnarle e prevenire i loop.

Esistono tre possibili meccanismi di segnalazione:

1. Label Distribution Protocol (LDP)

Protocollo hop by hop, segue il routing IP e quindi non supporta il *Traffic Engineering*.

2. Resource Reservation Protocol (RSVP-TE)

meccanismo più complesso che supporta nativamente il constrained - based routin e route esplicite. Supporta il *Traffic Engineering*. Permette anche di definire delle route di protezione che intervengono se una delle route diventa non raggiungibile semplicemente cambiando la label nel router di bordo.

3. Constrained Routing LDP

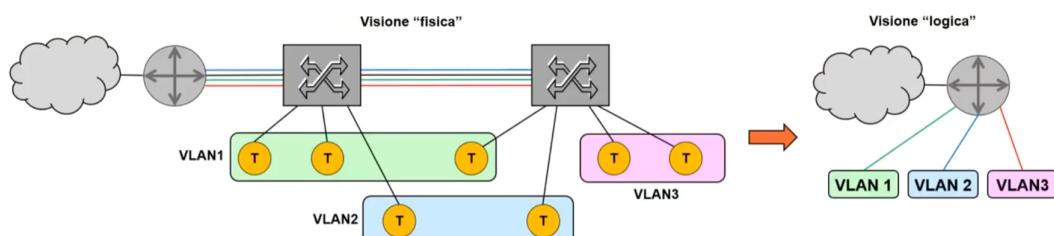
estende LDP per supportare il constrained - based routing e le route esplicite.

MPLS abilita il **constrained - based routing**, ovvero l'instradamento sulla base di certi vincoli come banda, richieste amministrative, ecc. I cammini possono essere determinati con calcoli offline (*con una certa ottimizzazione globale, possibile solo se si conoscono a priori le richieste di instaurazione di LSP*) o online (*richiesta di instaurazione di cammini in modo dinamico*).

RETI PRIVATE VIRTUALI (VPN)

Quando si parla di reti private virtuali solitamente si hanno delle porzioni di rete che hanno lo stesso spazio di indirizzamento e che si vogliono interconnettere creando la percezione che queste porzioni siano di fatto un'unica rete. Esistono reti private virtuali di livello 2 (*VLAN Ethernet, MPLS Virtual Private LAN Service*) e di livello 3 (*IP Tunnelling*).

VLAN ETHERNET



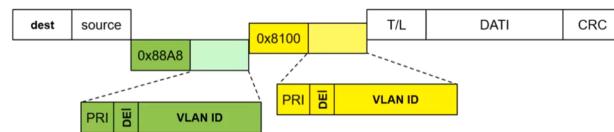
Le Virtual LAN sono nate per avere maggiore flessibilità in reti di campus / edificio e permettono di creare delle LAN Virtuali usando meno switch permettendo anche di estendere una VLAN su più switch (*nell'esempio abbiamo tre VLAN su due switch fisici*).

Questo viene fatto per mezzo di quattro byte aggiuntivi nella trama Ethernet, ovvero il cosiddetto **VLAN Tag** che contiene:

- **VLAN ID (12 bit)**: identificativo univoco della VLAN.
- **PRI (3 bit)**: bit di priorità.
- **DEI (1 bit)**: bit di *discard eligibility*, indica se una trama può o meno essere candidata allo scarto in condizioni di stress.

Ovviamente per poter fare tutto ciò, le porte degli switch devono essere **VLAN - Aware**, cioè devono poter distinguere il traffico di VLAN differenti. Per i terminali questo non è necessario. Tutto questo permette una grande riconfigurabilità e una segregazione del traffico e si ha chiaramente una maggiore sicurezza.

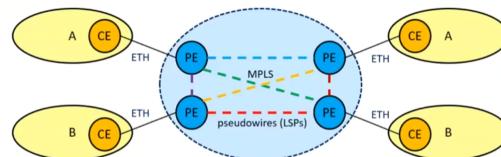
Esiste anche il **provider bridging** dove le VLAN configurate dagli utenti (*giallo*) sono raggruppate in VLAN gestite da un provider (*verde*).



MPLS VIRTUAL PRIVATE LAN SERVICE

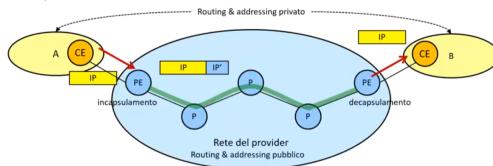
Una soluzione per creare una VPN di livello due estesa geograficamente (*WAN*) che risulta utile in caso di reti di datacenter. La rete MPLS si comporta, dal punto di vista logico, come un insieme di bridge interconnessi.

Vengono trasportate le trame Ethernet all'interno di pacchetti MPLS (*i CE sono apparati di livello 2*). La rete MPLS effettua il forwarding delle trame Ethernet ricevute dai PE su **porte virtuali** che sono associate a **pseudo - wires (LSPs)** come se fossero effettivamente dei collegamenti di livello 2.



IP TUNNELLING

L'IP Tunnelling permette di interconnettere delle reti che hanno lo stesso routing e addressing privato creando di fatto una VPN di livello 3 per mezzo di un meccanismo di **tunnelling** che prevede un **incapsulamento** dei pacchetti IP (assegnando l'IP dei PE) per permettere il trasporto in maniera trasparente da un punto ad un altro della rete. Questo risulta utile per la creazione di **reti IP private** distribuite geograficamente sfruttando una rete IP pubblica. Tuttavia MPLS è più efficiente, ma IP Tunnelling risulta più vantaggioso in situazioni di **interlavoro**, ovvero quando si hanno siti distribuiti su più stati, poiché ci sono più reti IP interconnesse e gestite da diversi provider, mentre le reti MPLS sono solitamente gestite da un solo provider.



NETWORKING DEVICES / ADVANCED NETWORKING TECHNOLOGIES

I dispositivi di networking sono i *mattoni* necessari a costruire una moderna rete di telecomunicazioni.

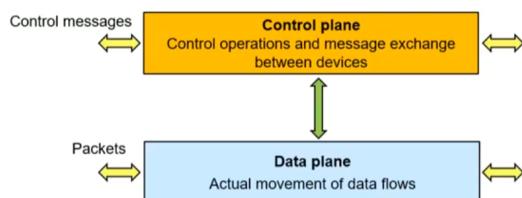
Abbiamo

1. **Dispositivi per *forwarding* e *routing*** come hub, switch, bridge, router.
2. **Dispositivi che garantiscono altre *funzionalità di rete*** come firewall, intrusion detection system, anti DDoS, load balancer.

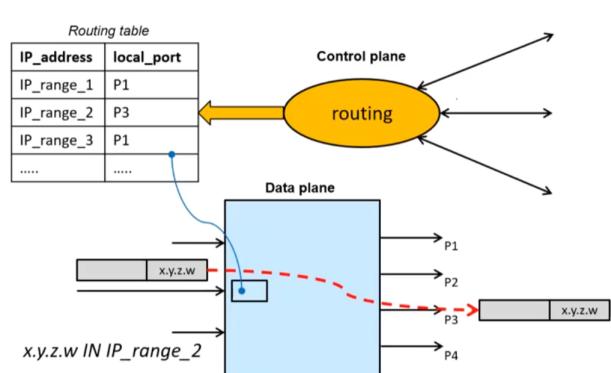
Tutti i dispositivi di rete hanno funzionalità che si dividono in:

1. **Piano Dati**: gestisce i singoli pacchetti localmente, porta a prendere decisioni locali (*ad esempio su quale interfaccia mandare il pacchetto*). Chiaramente queste decisioni devono essere prese molto velocemente.
2. **Piano di Controllo**: gestisce il flusso di controllo ed effettua le operazioni che fanno in modo che il dispositivo si comporti come deve considerando una visione globale della rete. Ad esempio gestisce i percorsi che devono seguire i pacchetti tra sorgente e destinazione, decide quali pacchetti vanno filtrati.

In genere, il piano di controllo prevede delle operazioni più complesse rispetto al piano dati.

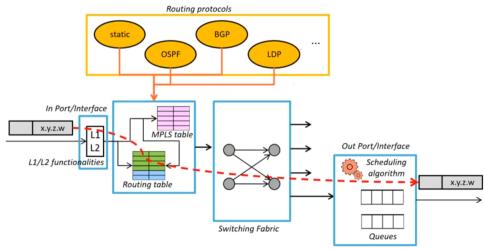


ROUTER



Il **piano di controllo** si occupa di popolare la tabella di routing (*che si trova nel piano dati*) per mezzo di protocolli di routing distribuiti.

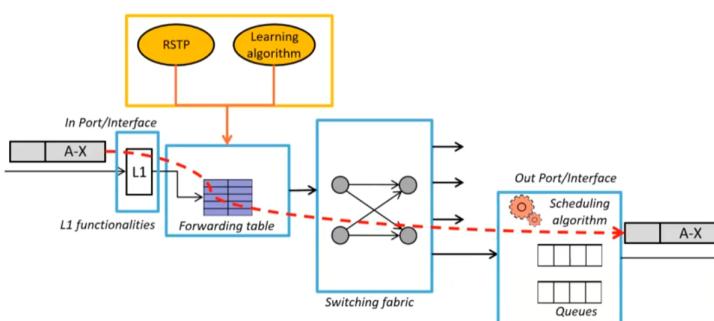
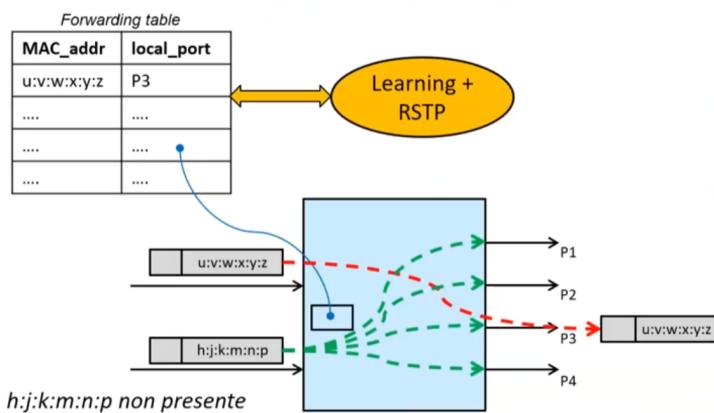
Quando arriva un pacchetto si cerca un match nella tabella di routing all'interno del **piano dati** e, se lo si trova, si inoltra il pacchetto sulla corrispondente interfaccia, altrimenti il pacchetto è scartato. Vengono fatte anche delle operazioni all'header (*come decrementare il TTL*).



A livello di **piano di controllo** spesso si hanno più algoritmi di routing che comunicano (*BGP, LDP, ...*). A livello di **piano dati** ci sono più componenti che interagiscono fra loro come lo **switching fabric** (*hardware high-performance che interconnette le interfacce in/out*) e le **code e scheduling di algoritmi** usate per gestire traffici con diverse richieste di QoS.

Il processo di *forwarding* prevede diverse fasi come *matching, longest prefix, preferences, metrics, equal cost multipath* (*distribuisce il traffico tra più rotte*). Generalmente la scelta dell'interfaccia può essere soggetta a più regole (*ad esempio se vogliamo privilegiare una rotta statica*), solitamente si sceglie quella a costo minore.

SWITCH



Lo switch presenta, sul piano dati, la **forwarding table** che contiene i vari MAC Address e le relative porte (*lo switch lavora a livello 2*). Sul piano di controllo abbiamo il meccanismo di **learning e forwarding** che usa il MAC address di sorgente per capire da dove arriva una trama e lo memorizza nella tabella (*è usato quindi per la popolazione della forwarding table*) e il meccanismo di **Rapid Spanning Tree Protocol** che è un protocollo distribuito che evita loop nel forwarding disattivando alcune porte creando così un albero logico del traffico.

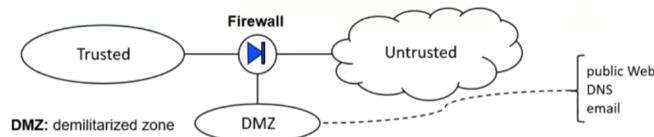
A livello di switch, a differenza del router, non si parla più di longest prefix match, ma solo di exact match trattandosi di indirizzi MAC.

Dunque, quando arriva una trama, se esiste un match esatto nella forwarding table, si inoltra la trama sulla relativa porta, altrimenti questa viene inviata in broadcast su tutte le porte.

MIDDLEBOXES

Abbiamo anche dei dispositivi definiti *middleboxes* che possono offrire sia funzionalità di piano dati (*Firewall, IDS, Anti DDoS, Load Balancer, NAT*) che di piano di controllo (*Authentication, DHCP, DNS, CDN*). Solitamente i middlebox relativi al piano dati necessitano di hardware specializzato per questioni di performance.

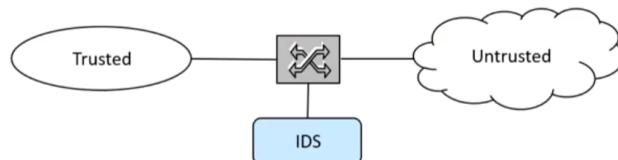
FIREWALL



Il firewall è un dispositivo *in line* (ovvero inserito nella rete tra una porzione *trusted* e una porzione *untrusted*) che implementa un **set di regole** sul traffico di rete che decideranno se lasciarlo passare o meno. Spesso si hanno anche delle **zone demilitarizzate (DMZ)** che sono delle parti della rete *trusted* che però mantiene servizi pubblici.

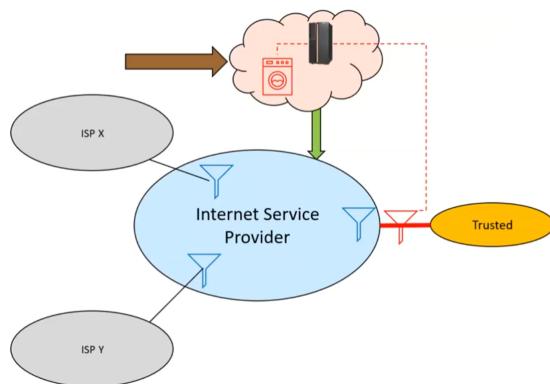
Il firewall è un dispositivo **stateful** e quindi tiene traccia dello stato del sistema e quindi sa in che stato sono le connessioni della rete (*ad esempio potrebbero accettare SYNACK solo dalle porte TCP che hanno ricevuto un SYN*). Essendo un dispositivo in line inoltre deve processare queste regole molto rapidamente e quindi spesso si utilizza dell'hardware specializzato per fare ciò.

IDS



L'Intrusion Detection System svolge un lavoro simile al firewall ma fa delle analisi molto più complesse sul traffico dati. Inoltre, questo non è un dispositivo in line (*anche perché le operazioni complesse che fa creerebbero un delay non indifferente se fosse in line*) e le sue operazioni mirano a identificare possibili attacchi alla rete.

ANTI DDOS



Un attacco DDoS è un attacco che cerca di interrompere l'accesso ad un servizio offerto andando ad esaurire le risorse della vittima. Esistono due tipi di sistemi Anti DDoS:

1. **ISP Based:** offerto nativamente dall'ISP (*soltanamente sotto un certo pagamento*).
2. **Cloud Based:** tutto il traffico in ingresso è deviato al cloud provider che offre il servizio e pulisce il traffico.

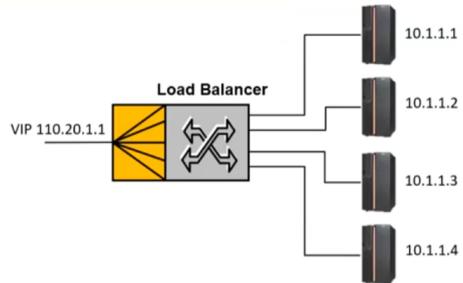
LOAD BALANCER

Il load balancing è un problema quando si ha a che fare con traffico proveniente da milioni di utenti che deve essere gestito.

Si hanno tre possibilità:

1. **Soluzioni Network - Based:** si usano tecniche che sfruttano la replica dei contenuti (*caching, CDN*) offerte da provider esterni o tecniche basate sul DNS che instrada gli utenti alla cache più vicina.
2. **Soluzioni Application - Based:** ad esempio reverse proxy.

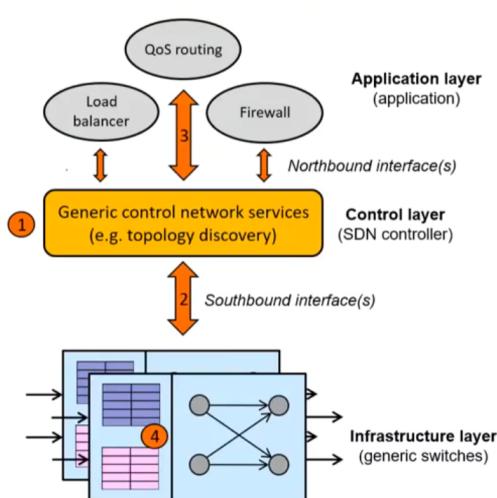
3. Soluzioni Hardware - Based: si usano dei load balancer middlebox (*tipici nei data center*).



I load balancer usano un IP virtuale che viene fornito agli utenti e smista le varie connessioni tra più risorse hardware. Un aspetto fondamentale è che va garantita la **persistenza delle sessioni** (*il traffico delle stesse sessioni deve essere gestito dagli stessi server*) e sono necessari **algoritmi di load balancing** che possono essere rigidi (*round robin*) o adattivi (*in base al carico di ciascun server, in base al tempo di risposta di questi*).

SOFTWARE DEFINED NETWORKS (SDN)

Abbiamo visto come i dispositivi visti fino ad ora hanno sempre un *piano di controllo* che può essere distribuito e un *piano dati* con tabelle e interconnessioni tra porte. Questi due piani sono logicamente separati (*anche se interagiscono*) ma sono fisicamente collocati.



In SDN, che è un paradigma nato nel 2008, i due piani vengono separati anche fisicamente in modo che il comportamento della rete sia **programmabile**.

Il piano di controllo in particolare è implementato a livello centralizzato in un **SDN Controller** che è implementato usando hardware di comodità e usa un *Network Operating System* che gestisce tutta la logica di controllo della rete. Il piano dati invece è implementato da dei nodi distribuiti (*chiamati tipicamente switch*) che implementano delle **tabelle generiche** (*flow tables*) popolate dal SDN Controller (*seguono il paradigma match + action*).

Il piano di controllo comunica col piano dati mediante delle **Southbound Interfaces (SBI)** remote. Inoltre, il piano di controllo comunica con le varie applicazioni mediante delle **Northbound Interfaces (NBI)**.

Questo permette di evitare algoritmi distribuiti in cui i router devono scambiarsi messaggi e ogni router si costruisce la topologia, ma di avere una visibilità globale direttamente nel piano di controllo.

I messaggi del controller sono tradotti in **regole di flow** e inserite nelle *flow tables* (*il flow è rappresentato da gruppi di pacchetti che soddisfanno gli stessi requisiti*). Queste *flow tables* sono concatenate in una pipeline (*ad esempio per prioritizzare dei pacchetti*) e solo la prima è obbligatoria. Si hanno anche una **group table** usata per la comunicazione multicast e una **meter table** per le statistiche.

Le *flow table* sono esplorate in ordine crescente rispetto al loro indice (*anche per evitare loop*).

Questo tipo di soluzione è stata adottata perché ci si è resi conto che l'approccio stacked introdotto alla nascita di Internet ai tempi d'oggi introduce dei delay che possono essere risolti andando a controllare allo stesso tempo più campi di più header (*dal livello 2 al livello 4*).

In queste tabelle sono riportate le varie regole e delle azioni che possono essere ad esempio di forwarding presso un'altra porta o al controller, di drop del pacchetto, di modifica dell'header del pacchetto. Ogni pacchetto ha dei *metadati* che contengono le istruzioni su cosa va fatto sul determinato pacchetto.

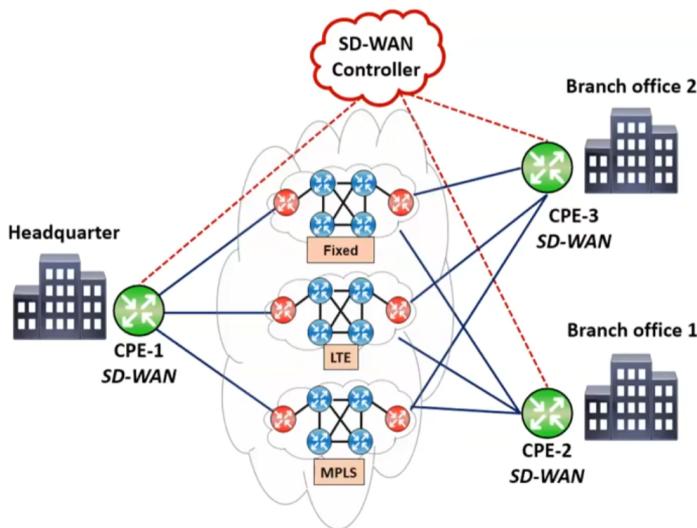
Le flow rules sono inserite nelle tabelle di flusso in due modi:

1. **Configurazione Proattiva:** le flow table sono completamente configurate dal controller al startup dello switch, tutte le entry sono statiche e non modificabili.
2. **Configurazione Reattiva:** le flow table sono configurate a runtime (*all'inizio sono vuote*). Quando un pacchetto non matcha nessuna regola, viene inviato al controller perché si occupi dell'instaurazione delle regole necessarie.

La soluzione che si adotta tipicamente è un ibrido tra le due, ovvero vengono instaurate delle regole allo startup che coprano il più possibile i flow conosciuti, mentre per i flow sconosciuti le regole vengono istanziate a runtime. Ovviamente è difficile configurare le flow table, bisogna garantire che il controller non sia bloccato da troppe richieste da pacchetti sconosciuti.

APPLICARE PRINCIPI SDN ALLE WAN

Per ridurre i costi nella connettività WAN si sono adottate delle tecnologie SDN andando a creare le SD-WAN. L'idea di base è di stipulare diversi contratti di connettività generalizzata su diverse tecnologie (*xDSL, 4G, PON, ecc*) e usarle in modo intelligente per garantire un certo QoS.



le decisioni giuste a garantire la qualità del servizio target.

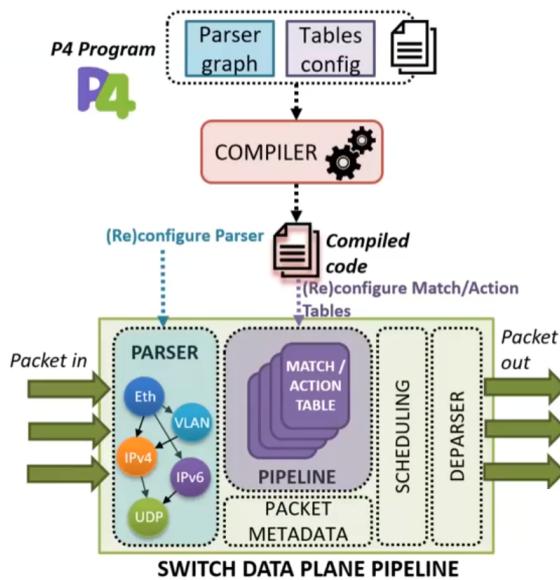
Si hanno delle SD-WAN Box installate in ogni sede (*Customer Premises Equipment*) e un SD-WAN controller che controlla il traffico e che indica ai vari CPE come sono organizzate le varie connessioni.

Questa soluzione è molto più economica rispetto alle WAN MPLS (*ma può essere usata insieme a MPLS*). Chiaramente le varie connessioni vanno monitorate per poter prendere

DATA PLANE PROGRAMMING

Soluzioni SDN come OpenFlow (*o simili*) è possibile programmare il comportamento della rete, ma non è possibile modificare il piano dati in maniera flessibile. Ad esempio se si vuole aggiungere un nuovo **matching field** o si vuole supportare un nuovo **protocollo**, l'hardware va cambiato.

Data Plane Programming vuole rendere, oltre al piano di controllo, anche il piano dati programmabile e quindi flessibile. Dunque la pipeline può essere programmabile e si può modificare la struttura delle tabelle di flusso.



Uno di questi protocolli è **PISA (Protocol Independent Switch Architecture)**

Notiamo che nello switch questa volta abbiamo un **parser** che può essere programmato e che quindi permette di definire come deve essere trattato il pacchetto. Si hanno anche qui un insieme di *match / action tables* che però sono programmabili in modo flessibile. Il comportamento della pipeline può essere definito tramite un file scritto in linguaggio P4 che viene compilato e l'eseguibile compilato riconfigura il parser e le tabelle *match / action*.

NETWORK FUNCTION VIRTUALIZATION

Network Function Virtualization, a differenza di SDN che è nato in ambito accademico, è nato in ambito industriale. NFV cerca di disaccoppiare l'hardware dal software nelle implementazioni delle funzioni di rete (*ovvero le funzioni implementate dai middleboxes*). Quello che si vuole fare è usare del generic purpose hardware per eseguire software specializzati che permettono di avere le funzionalità di rete virtualizzate. Questo chiaramente permette una maggiore flessibilità in quanto queste funzioni di rete si possono deployare ovunque e una maggiore scalabilità in quanto è semplice fare lo scale up di questo tipo di soluzioni. Inoltre, l'hardware spesso diventa obsoleto molto velocemente, quindi virtualizzare il tutto aiuta anche sotto questo punto di vista.

Questo approccio funziona molto bene per la virtualizzazione delle funzioni del piano del controllo, ma meno bene per le funzioni del piano dati poiché queste richiedono una **velocità di processamento dei pacchetti molto alta**, mentre le funzioni del piano di controllo generalmente richiedono meno velocità computazionale (*anche se le operazioni sono più complesse di quelle del piano dati*). Mentre SDN è il primo passo verso la virtualizzazione delle rei, NFV è il primo passo verso la **cloudificazione** della rete.

QUALITY OF SERVICE

La qualità del servizio (QoS) è un **indice di qualità** che misura il livello di servizio rispetto alle attese dell'utente. La QoS è associata ai servizi e ai rispettivi flussi di traffico ed è strettamente dipendente da banda disponibile, ritardi, packet dropping, blocking probability, setup delay, ecc.

La QoS è legata ma si distingue dalla **Quality of Experience (QoE)** che è un indice che misura in termini **soggettivi** il valore del servizio offerto all'utente, mentre la QoS è **oggettiva**. Possiamo inoltre definire la QoS in modo **assoluto** definendo dei valori che devono essere rispettati dall'insieme dei parametri prestazionali (es. *ritardo end-to-end < 20 ms*) o **relativo** definendo la modalità di trattamento di una classe di traffico rispetto ad un'altra.

Sappiamo che Internet è **best - effor** e non da quindi nessuna garanzia, ma la richiesta di qualità nelle reti IP è cresciuta negli ultimi decenni e per questo sono state sviluppate soluzioni per garantirla (*MPLS, IP diffserv, IP intserv ...*).

Il traffico in rete è di tipo **bursty** il che significa che si hanno degli istanti di tempo in cui non viene

invia nulla e istanti in cui il traffico viene inviato tutto insieme: questo aspetto deve essere tenuto in considerazione quando si fanno valutazioni di QoS.

Possiamo definire due tipologie di servizi:

1. **Servizi real time**: molto sensibili al ritardo, ma meno alla perdita (*VoIP*)
2. **Servizi elastici**: molto sensibili alla perdita, ma meno al ritardo. Solitamente richiedono ACK di corretta consegna (*web browsing*).

Ovviamente diverse tipologie di servizi richiedono diversi requisiti QoS e quindi sono necessari meccanismi che li garantiscano.

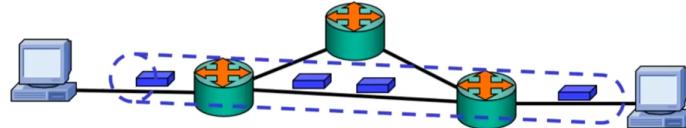
STRUMENTI PER LA GARANZIA DELLA QOS

Chiaramente per garantire una certa QoS sono necessari alcuni strumenti quali

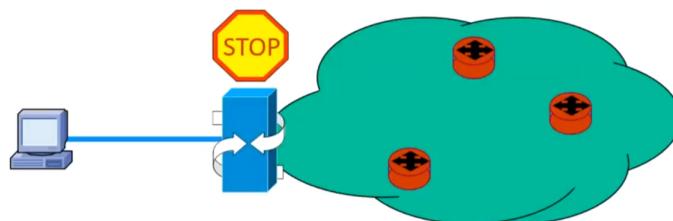
1. Strumenti di **identificazione dei flussi** e/o di **tipologia di traffico** (es. *label*, *VLAN Tag*)



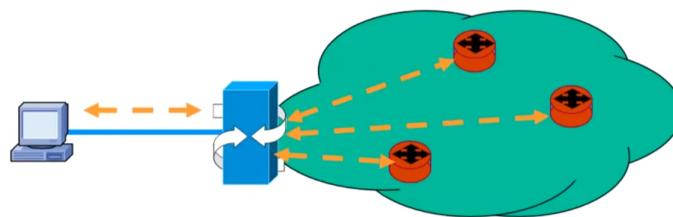
2. Strumenti di **traffic engineering** per fissare i cammini in rete (es. *MPLS*)



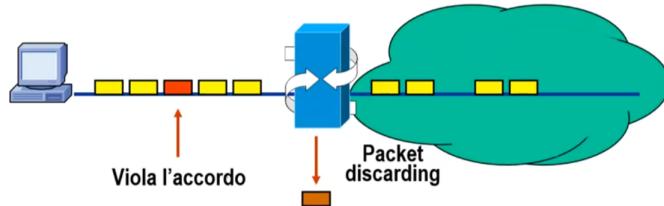
3. Meccanismi di **Call Admission Control (CAC)** che possano rifiutare la richiesta se non sono disponibili sufficienti risorse per garantirla o per continuare a garantire le altre già ammesse alla qualità definita.



4. Meccanismi di **segnalazione delle risorse di rete** (es. *banda disponibile sui cammini*) che risultano utili per prendere decisioni migliori in termini di CAC (es. *traffic engineering database popolato estendendo le funzionalità dei protocolli di routing*).



5. Meccanismi di **regolazione del traffico** che controllino che il traffico immesso nella rete sia conforme alle risorse disponibili ed eventuali accordi stipulati e che prendano provvedimenti in caso di violazione.



6. Strumenti per prioritizzare il traffico in uscita nei dispositivi di rete (**tecniche di scheduling**).

7. **Over - Provisioning**, ovvero dimensionamento delle risorse di rete ben oltre quanto mediamente necessario in modo da evitare contese nell'utilizzo delle risorse. Chiaramente questo ha un grosso costo rispetto ad esempio usare dei meccanismi per aumentare mediamente l'occupazione delle risorse.

REGOLAZIONE DEL TRAFFICO

Solitamente un provider di servizi IP stabilisce col cliente due contratti, un **Service Level Agreement (SLA)** e un **Traffic Conditioning Agreement (TCA)**.

Il SLA specifica la QoS che il provider si impegna a garantire per il traffico relativo a una tipologia di servizio specifica ed è definito sulla base di vari **parametri misurabili (metriche)** come ritardi end-to-end, throughput, tasso di perdita, ecc.

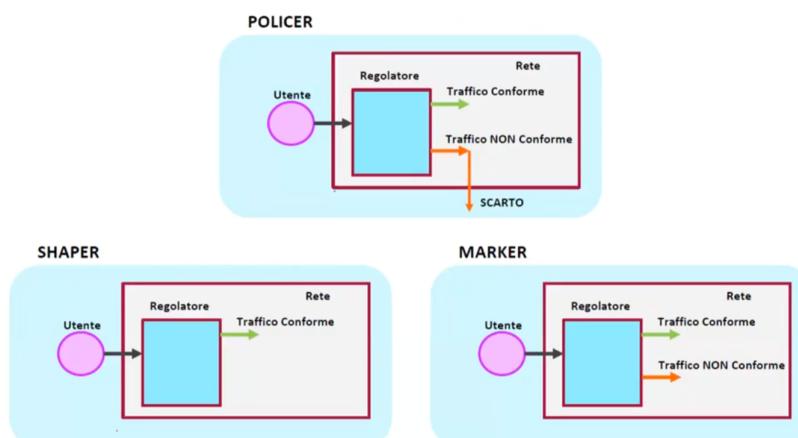
Include vari **Service Level Objective (SLO)** che indicano la QoS da garantire nell'unità di tempo e sono definiti per mezzo dei parametri misurabili.

Chiaramente l'impegno del provider sulla garanzia dello SLA non può prescindere dalla quantità di traffico generata dal cliente e quindi il TCA specifica il **profilo di traffico**. Esso è formato da parametri caratterizzanti il traffico offerto dal cliente per cui il provider assicura l'impegno dei onorare il SLA. Il traffico conforme al TCA si chiama **traffico in (o conforme)** mentre quello non conforme si chiama **traffico out (o non conforme)**.

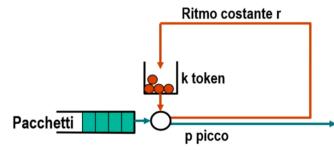
I parametri del TCA sono ad esempio il tasso di picco (*peak rate*), il tasso medio (*average rate*), la massima lunghezza dei burst, la massima/minima lunghezza dei pacchetti.

Ovviamente l'accettazione del traffico non conforme è rischiosa in quanto potrebbe consumare risorse in misura tale da compromettere la capacità di garantire gli SLA stipulati (*anche con altri clienti*). Il traffico non conforme può però essere trattato in diversi modi come **policing** (*viene scartato*), **shaping** (*viene ritardato in modo da attribuire un comportamento conforme al TCA*) o **marking** (*viene contrassegnato in modo tale da essere riconosciuto ed eliminato in caso di necessità*).

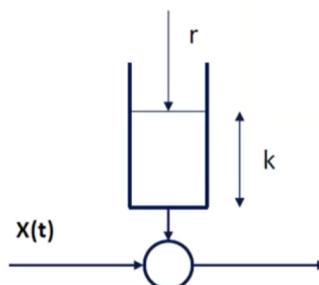
Il traffico viene esaminato da un regolatore che distingue quello conforme da quello non conforme. Il trattamento del traffico non conforme distingue i diversi tipi di regolatore che usa degli algoritmi specifici per controllare i parametri definiti nel TCA.



TOKEN BUCKET

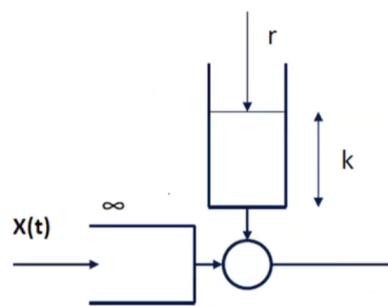


Un primo algoritmo implementato dai regolatori per controllare i parametri del TCA è il token bucket. Questo algoritmo controlla la velocità di picco p (bit/s), la velocità media b (bit/s) e la lunghezza del burst L (s).



POLICER

Il *token bucket policing regulator (policer)* ha un serbatoio (**bucket**) di **token** con dimensione massima di k unità di traffico (*token bucket size, misurata in bit, byte o pacchetti*). Il bucket è incrementato a ritmo costante ogni $1/r$ con r **token rate**. Il dispositivo ammette il passaggio di un'unità di traffico solo se il bucket contiene almeno un token e, in tal caso, il bucket viene decrementato di un token. Se il bucket è vuoto, il traffico viene scartato.



SHAPER

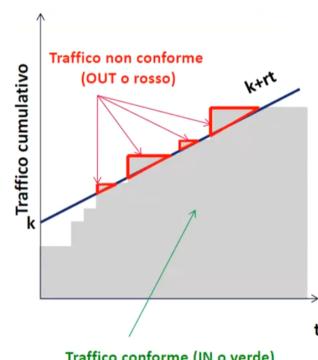
Token bucket può essere utilizzato anche per implementare uno shaper. Il bucket opera allo stesso modo del policer. In questo caso un'unità di traffico passa attraverso il regolatore se al suo arrivo il bucket ha almeno un token e se il **buffer di ingresso** (*di dimensione infinita*) è vuoto. Se il buffer non è vuoto e/o il bucket non ha più token, l'unità di traffico si accoda nel buffer di ingresso. Quando il buffer non è vuoto, un'unità di traffico viene prelevata ed inoltrata non appena si ha un token disponibile.

TRAFFICO BURSTY

Come abbiamo detto in precedenza, il traffico in rete è di tipo bursty e la presenza del bucket di token consente al traffico che viene generato dalla sorgente di essere ammesso in rete ad una velocità maggiore della velocità media (*alla velocità di picco p*). I token bucket cercano di mantenere la *burstyness* calcolando la durata massima dei burst come $L = k/(p - r)$.

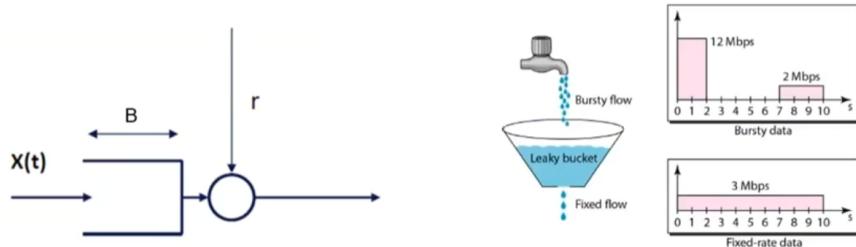
FUNZIONE DI VINCOLO

I regolatori policer e shaper implementano una **funzione di vincolo** definita da una retta $k + rt$ che rappresenta il massimo numero di unità traffico corrispondenti al traffico conforme che sono state lasciate passare. In un certo tempo \tilde{t} , il regolatore ha lasciato passare al più $k + r\tilde{t}$ unità di traffico conforme. Uno shaper, quando incontra del traffico out, si allignerà alla funzione di vincolo invece che assumere valori sopra di essa.



LEAKY BUCKET

Un algoritmo alternativo al token bucket è il leaky bucket che non ha un serbatoio dei token, ma viene generato comunque un token ogni $1/r$. Il buffer non è più infinito, ma ha dimensione B : oltre questa dimensione massima il traffico viene scartato. Poiché i token non si accumulano, un traffico burst viene reso *smooth* e non possiamo avere velocità d'uscita superiore a r che è quindi anche la velocità di picco ($r = p$).



ALLOCAZIONE DELLE RISORSE

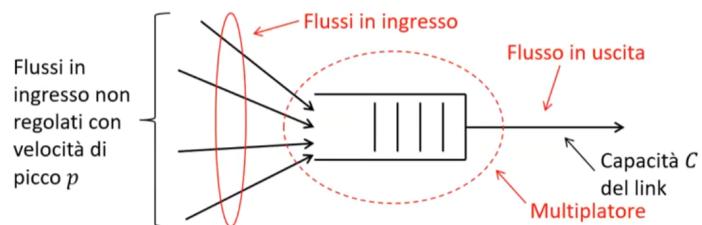
Le tecniche di policing, shaping e marking permettono di migliorare l'allocazione delle risorse in rete. Quando però diversi flussi sono multiplati in un unico flusso le risorse devono essere allocate in modo intelligente.

Esistono due tipologie di allocazione:

1. **Allocazione deterministica:** le risorse sono suddivise staticamente tra i differenti flussi in ingresso e non è prevista perdita per i pacchetti dei flussi. Si usano **allocazione al picco** e **allocazione mediante Dual Leaky Bucket**.
2. **Allocazione statistica:** le risorse sono suddivise in modo dinamico tra i flussi secondo criteri statistici, rendendo possibile la perdita di pacchetti che deve essere controllata.

ALLOCAZIONE AL PICCO

Viene utilizzata quando il traffico relativo ai flussi in ingresso non è stato regolato da alcun regolatore. A ogni flusso viene assegnata una porzione di capacità C del link in uscita pari alla velocità di picco p .



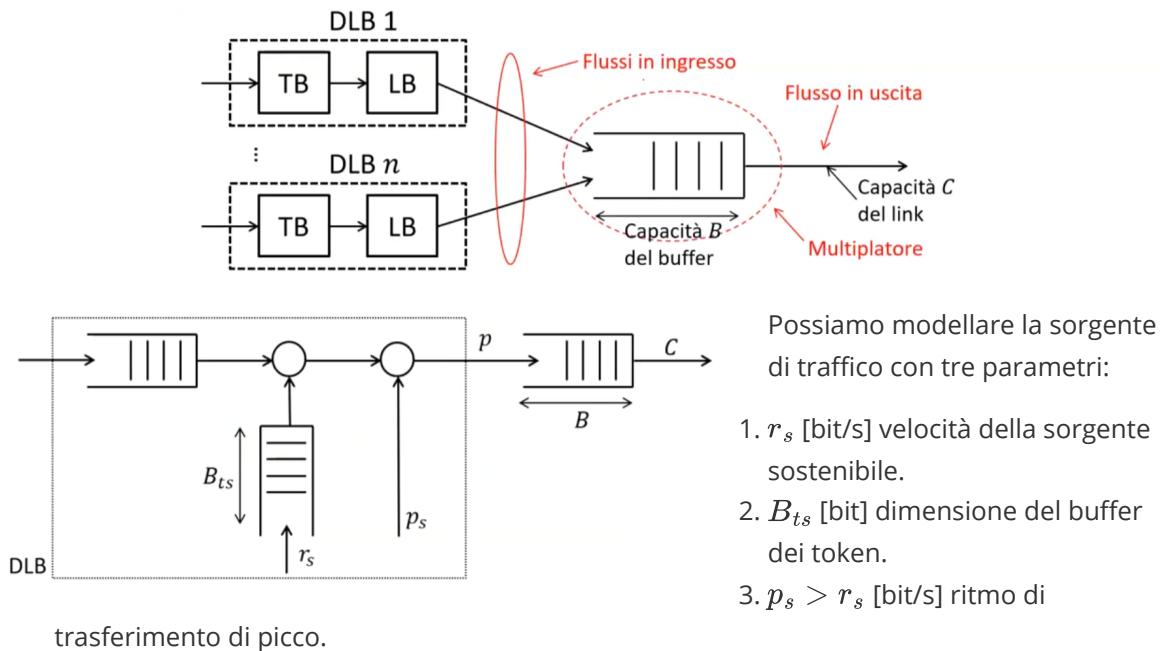
Supponendo di avere flussi in ingresso omogenei (stessa velocità di picco p), il numero massimo di flussi che possono essere multiplati senza perdita è $n = \frac{C}{p}$. Se $b_m \leq p$ è la velocità media di ognuno dei flussi, possiamo definire **efficienza** della strategia di allocazione come

$$\eta = \frac{n \cdot b_m}{C} = \frac{b_m}{p}$$

L'allocazione al picco allora le risorse sulla base del caso peggiore e, più è grande la differenza tra velocità media e velocità di picco, più bassa sarà l'efficienza. Di contro però garantisce l'assenza di perdita di pacchetti in ogni condizione.

ALLOCAZIONE DUAL LEAKY BUCKET

Questo tipo di allocazione è possibile quando il traffico dei flussi in ingresso è stato regolato da una cascata ***token bucket + leaky bucket***.



Il primo elemento (*TB*) regola la velocità d'uscita r_s permettendo una certa tolleranza (*che dipende da B_{ts}*). Il secondo elemento (*LB*) regola la velocità di picco $p_s \leq p$ (*p* velocità di linea).

Questa soluzione permette di ridurre la durata dei burst ad un valore massimo prefissato e permette di effettuare un'allocazione deterministica senza perdita efficiente nel caso in cui vengano dimensionati correttamente i parametri B e C , ovvero la dimensione del buffer del multiplatore e la capacità d'uscita del multiplatore.

La presenza del DLB impone un limite al tempo massimo in cui è possibile inviare pacchetti alla velocità di picco p_s (*durata massima del burst*) definita come $T_{picco} = L = \frac{B_{ts}}{p_s - r_s}$

Si assume che si alloca una stessa porzione di banda e buffer (c, b) a ogni flusso che sia una frazione di (C, B) .

Sia n il numero massimo di flussi nel sistema, abbiamo che $B = nb$ e $C = nc$.

Si devono rispettare due vincoli per evitare perdite:

1. Un vincolo di ritardo massimo imposto. Per farlo dobbiamo dimensionare in modo opportuno B e C .

$$D_{max} = \frac{B}{C} = \frac{nb}{nc} = \frac{b}{c} \text{ [s]}$$

2. La porzione di buffer b deve essere congrua per ogni flusso

$$b = (p_s - c)T_{picco} \text{ [bit]}$$

Moltiplicando entrambi i termini per n abbiamo

$$nb = n(p_s - c)T_{picco}$$

$$nb = (np_s - nc)T_{picco}$$

$$B = (np_s - C)T_{picco}$$

A questo punto possiamo impostare il sistema $\begin{cases} B = C \cdot D_{max} \\ B = (np_s - C)T_{picco} \end{cases}$

$$\text{E possiamo calcolare } n = \frac{C}{p_s} \left(1 + \frac{D_{max}}{T_{picco}}\right) = \frac{c}{p_s} \left(1 + \frac{D_{max}(p_s - r_s)}{B_{ts}}\right)$$

Notiamo quindi che variando opportunamente p_s , r e B_{ts} delle sorgenti possiamo ottenere valori maggiori o minori di n .

CONFRONTO

Abbiamo visto che, nel caso di allocazione al picco abbiamo $n_p = \frac{C}{p}$ mentre nel caso di allocazione DLB abbiamo $n_{DLB} = \frac{C}{p_s} \left(1 + \frac{D_{max}}{T_{picco}}\right)$

Essendo $p \geq p_s$ abbiamo che $n_{DLB} \geq n_p$.

SCHEDULING

Le tecniche di scheduling permettono di suddividere la banda in uscita nelle interfacce dei router che viene condivisa tra pacchetti di diverse code in ingresso. Lo scheduler determina in che modo la banda debba essere suddivisa per mezzo di diverse possibili strategie.

Una prima strategia è il **Time Division Multiplexing** che assegna degli slot di tempo ad ogni coda. Questa tecnica tuttavia non consente il riutilizzo di risorse non usate da un flusso, ovvero se una coda non ha nulla da trasmettere di fatto *spreca* lo slot.

Un'altra tecnica è la **Round Robin (o Fair Queuing)** in cui la suddivisione dinamica della banda avviene permettendo ad ogni coda di trasmettere un pacchetto, se presente. Il concetto è simile alla TDM, ma in questo caso viene data la possibilità ad una coda di trasmettere solo se effettivamente deve farlo, in modo che non si sprechino slots. Lo *svantaggio* è che non sappiamo a priori chi trasmetterà come nel caso della TDM.

Esiste anche una versione **Weighted Fair Queuing** che permette ad ogni coda i di trasmettere al più k_i pacchetti in modo pseudo - ciclico. Permette di prioritizzare alcune code.

Tuttavia è possibile anche stabilire una **divisione dinamica preferenziale** dove ad ogni coda è assegnata una priorità e la trasmissione avviene per livello di priorità. Chiaramente le code con priorità bassa possono sperimentare forti ritardi, specialmente se la coda ad alta priorità trasmette pacchetti molto lunghi. Per risolvere questi problemi spesso è necessario frammentare i pacchetti.

Il problema è che se frammentiamo i pacchetti a livello IP solo il destinatario può ricombinarli e avere molti frammenti IP in rete porta ad enormi inefficienze. Per questo motivo vengono adottati meccanismi di frammentazione di livello 2 sulle connessioni lente dove la ricomposizione avviene in ricezione (*protocollo PPP (Point - to - Point Protocol)*).

CALL ADMISSION CONTROL

Il Call Admision Control è caratterizzato da un insieme di azioni intraprese da una rete durante la fase di **instaurazione o ri - negoziazione** di una connessione con l'obiettivo di stabilire se la richiesta di connessione possa essere o meno accettata. Quando viene effettuata questa operazioni ci si assicura che sul cammino seguito dal flusso in rete siano assegnati, su ogni link e nodo del cammino una porzione della capacità del link (**bandwidth assignment**) e una porzione del buffer nell'interfaccia di uscita del nodo (**buffer assignment**).

La quantità di banda e buffer dipende dai requisiti QoS.

Chiaramente l'entità che va ad effettuare la procedura di CAC deve conoscere la quantità di risorse necessaria ad evadere opportunamente la richiesta e la quantità di risorse usata attualmente su ogni link / nodo. Deve quindi verificare se esiste un cammino con tale disponibilità e prenotare le risorse in caso affermativo.

Le modalità di calcolo e prenotazione sono di tre tipi:

1. Modalità Centralizzata: il CAC viene effettuato da un server centrale.

In questo caso la segnalazione verso i nodi è semplificata e può essere determinato un cammino ottimale.

D'altro canto si hanno problemi di scalabilità e affidabilità e non è ben tollerato da IP se non per sistemi semplici (*l'instradamento di IP interferisce*).

2. Modalità Distribuita: ogni nodo della rete concorre al CAC.

Questo è un sistema più robusto e affidabile ma anche più complesso e che richiede un sistema di segnalazione per raccogliere la disponibilità degli altri nodi e per prenotare le risorse sul cammino (RSVP).

3. Modalità Mista: il CAC è eseguito solo ai nodi d'ingresso della rete.

I router di bordo ottengono periodicamente informazioni sullo stato delle risorse in rete e prendono le decisioni.

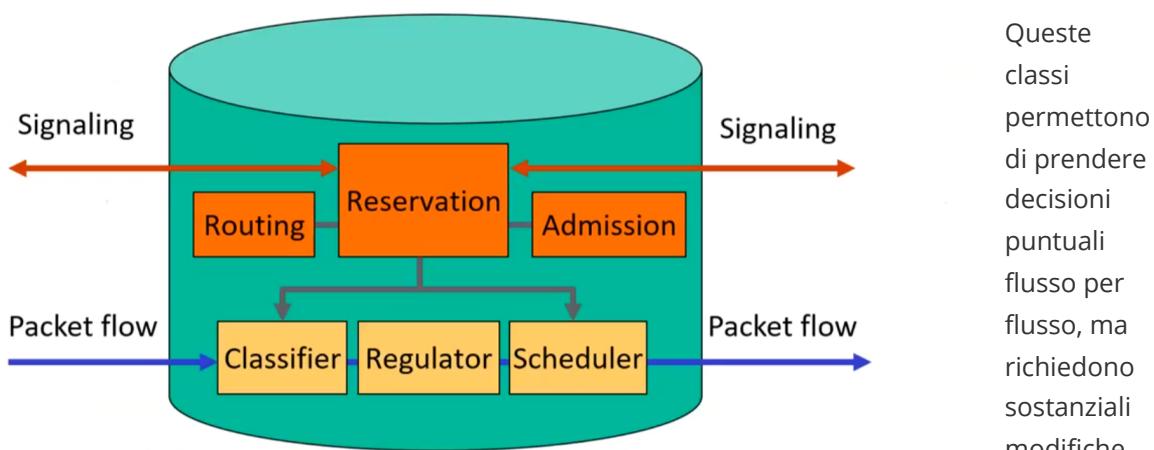
Questo sistema è meno complesso rispetto alla modalità distribuita e coinvolge meno nodi nella procedura di CAC, ma anche qui servono il sistema di segnalazione e di prenotazione delle risorse.

INTEGRATED SERVICES (INTSERV)

IntServ è il primo modello di servizio pensato per fornire QoS sulle reti IP e utilizza il protocollo **RSVP** per la prenotazione dinamica delle risorse di ogni flusso d'utente che richiede QoS. La QoS è definita in modo assoluto tramite un meccanismo di CAC ma col crescere delle necessità di garantire QoS a sempre più flussi ha mostrato grossi problemi di scalabilità.

Offre due classi di servizio oltre alla classe *best effort*:

- **Guaranteed Service:** emula il servizio a circuito con ritardi garantiti.
- **Controlled Load Service:** emula la modalità best effort ma in una rete non congestionata. Se si ha congestione, si differenzia il trattamento del traffico.



all'architettura dei router ed è un sistema complesso.

In particolare, i router devono avere, nel piano di controllo una parte che gestisca le **Reservation** e una parte di **Admission** che va ad effettuare le operazioni necessarie per capire se un flusso possa essere ammesso o meno. Nel piano dati invece sono richiesti un **Classificatore** (*identifica il*

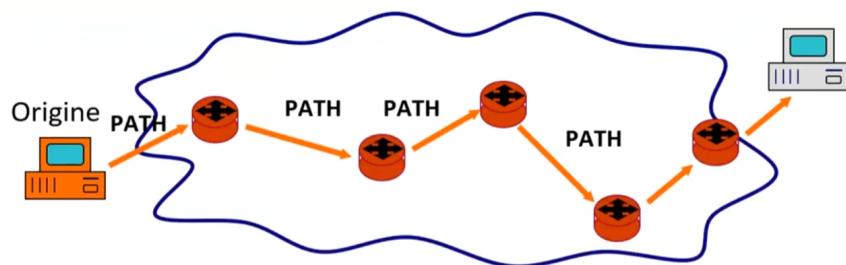
*flusso a cui appartiene il pacchetto in arrivo), un **Regolatore** (regolazione del traffico policing/shaping/marketing) e uno **Scheduler** (seleziona il pacchetto da inviare sul link in uscita).*

RESOURCE RESERVATION PROTOCOL (RSVP)

RSVP è un protocollo di livello 3 che viene encapsulato in un pacchetto IP con ID Protocol 46 e che viene usato per trasmettere ai router sul cammino scelto le informazioni di richiesta di risorse e QoS relative ad ogni flusso. I router valutano se le richieste possono essere accettate e in caso affermativo le memorizzano in modo da poterle soddisfare all'arrivo dei relativi flussi.

RSVP fa uso di diversi messaggi fra cui i più importanti sono **PATH** che fissa il cammino su cui deve essere effettuata la riservazione (*segue instradamento IP*) e **RESV**.

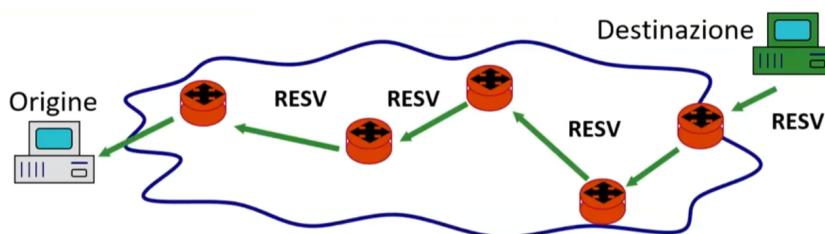
I messaggi *PATH* sono emessi periodicamente per consentire l'eventuale rivelazione di variazioni nell'instradamento.



Ogni router che riceve un messaggio di *PATH* mantiene un **path state** che include l'indirizzo del nodo precedente attraversato dal messaggio, i parametri relativi alle caratteristiche del flusso e l'interfaccia di ingresso e di uscita del messaggio *PATH*. Il **path state** è associato al **flusso** a cui si riferisce il messaggio *PATH*. Chiaramente, essendo i *PATH* relativi a vari flussi, se si devono gestire tanti flussi si ha un problema di scalabilità.

Il messaggio di *PATH* include due **oggetti** fondamentali:

- **TSPEC (Traffic SPECification)**: permette di informare i router sulle caratteristiche del flusso e contiene la descrizione del traffico generato dalla sorgente (*ad es. tramite i parametri del token bucket*). Non può essere modificato dai router.
- **ADSPEC (ADvertising SPECification)**: raccoglie informazioni preliminari riguardo la QoS sul cammino. Viene esaminato ed opportunamente modificato ad ogni hop con informazioni sulla QoS conseguibile sul cammino (*es. se ci sono router non RSVP-compliant, se alcune classi di servizio non sono supportate, ecc.*).



Quando il messaggio *PATH* arriva al destinatario, questo legge gli oggetti inclusi e vede quindi lo stato della rete riguardo la QoS e risponde con un messaggio **RESV** per effettuare le richieste di riservazione. Il messaggio *RESV* segue all'indietro il cammino fatto da *PATH* (*e quindi NON segue l'instradamento IP, fa source-based routing*) e, sulla base di TSPEC e ADSPEC, decide che richiesta di riservazione delle richieste fare.

Il messaggio di *RESV* include l'**oggetto** fondamentale **FLOWSPEC** che è composto dai campi **TSPEC** contenente le caratteristiche del flusso, eventualmente cambiate dal destinatario e **RSPEC** contenente i parametri QoS del tipo di servizio desiderato (*ad esempio specifica le garanzie che devono essere offerte al traffico nel caso di Guaranteed Service*).

La Call Admission viene quindi effettuata passo passo dai router che ricevono in input le informazioni sul flusso, conoscono l'occupazione delle proprie risorse dovuta a flussi già accettati e determinano le risorse da assegnare alla nuova chiamata. In caso esistano, accettano il nuovo flusso.

Nei router di bordo viene fatta solo la **regolazione** del traffico, mentre nei router interni si fanno anche **classificazione e scheduling**.

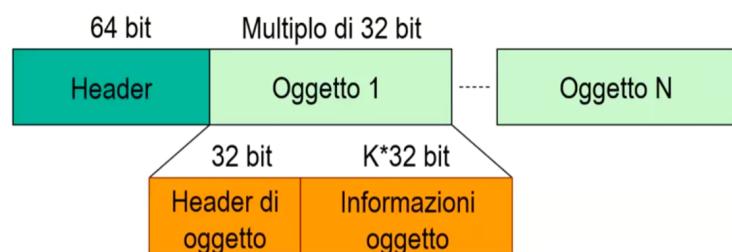
SOFT STATE

Abbiamo detto che i messaggi di *PATH* sono reinviati periodicamente, dunque il **path state** all'interno di ogni router è di fatto uno **soft state**, cioè ha una validità limitata nel tempo e controllata da un timer. Sorgente e destinazione si scambiano di continuo nuovi messaggi *PATH/RESV* e se l'instradamento cambia, il ricevente deve riservare esplicitamente risorse sul nuovo cammino.

Questo permette un recupero da situazioni di errore, una tolleranza alla perdita dei pacchetti di segnalazione e una buona adattabilità a cambiamenti del cammino tra sorgente e destinazione. Ovviamente però questo rende anche la segnalazione più pesante generando più traffico di segnalazione in rete.

Chiaramente RSVP richiede che venga limitata la congestione dei pacchetti di segnalazione riservando della banda per loro o usando dei meccanismi di priorità.

FORMATO MESSAGGI RSVP

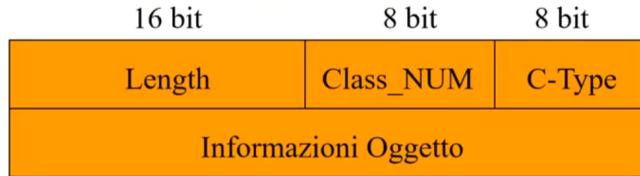


I messaggi RSVP hanno un header di 64 bit e contengono degli *oggetti*, ognuno col proprio header.

Vers	Flags	Msg Type	RSVP Checksum
Send_TTL	Reserved	RSVP Length	

L'header contiene informazioni circa la versione del protocollo utilizzato, il tipo di messaggio (*PATH, RESV, ...*), un checksum per la verifica dell'integrità del messaggio, la lunghezza e un TTL che, confrontato con quello dell'header IP, permette di individuare l'attraversamento di nodi non-RSVP-compliant.

L'header degli oggetti contiene invece la lunghezza, la classe (*TSPEC, ADSPEC, FLOWSPEC*) e il formato usato per l'oggetto (*IPv4, IPv6, MPLS*).



In IntServ, RSVP assegna le risorse in modo differente per le due classi di servizio supportate:

- **Guaranteed service** dove si emula un servizio a circuito con ritardi garantiti e senza perdita nei buffer di trasmissione.

In questo caso **TSPEC** definisce le caratteristiche del traffico e contiene i parametri del token bucket (*specificati dal senter ed eventualmente modificati dal receiver*) come bucket size k , token rate r e peak rate $p > r$.

ADSPEC contiene i parametri che permettono, lato ricezione, di stimare il ritardo end-to-end e la banda minima disponibile in rete.

RSPEC (in FLOWSPEC) contiene i dati relativi alla banda B da riservare per quel specifico flusso e allo *slack term* S che rappresenta il ritardo end-to-end tollerabile a cui è sottratto il ritardo end-to-end con la banda riservata.

Il ricevitore, sulla base dell'**ADSPEC** ricevuto (*dal messaggio PATH*), determina la banda B_j che i router devono assegnare al flusso j e lo *slack term* S .

Successivamente invia B_j e S come **RSPEC** (*messaggio RESV*).

Quando i router ricevono il messaggio, se S è positivo, possono ridurre a monte la banda B_j (*e di conseguenza S*) e aggiornare **RSPEC**, altrimenti non modificano i valori.

Se la banda B_j non è disponibile in un nodo sul percorso e non può essere ridotta perché S diverrebbe negativo, il flusso viene rifiutato.

In questo modo si garantisce una porzione di banda B_j riservata e che il ritardo end-to-end non superi un certo valore prefissato.

- **Controlled Load Service** dove si emula un servizio best effort in una rete non congestionata senza garanzie sui ritardi.

In questo caso la gestione dei nodi è più semplice e lascia ampio spazio alle implementazioni.

Anche in questo caso **TSPEC** contiene le caratteristiche del token bucket, mentre **ADSPEC** è privo di parametri QoS (*ed è opzionale*) e può includere informazioni sulla MTU lungo il percorso. Non si stima il ritardo.

FLOWSPEC non contiene **RSPEC** e ogni router decide la banda assegnata B_j necessaria a garantire il servizio.

PROBLEMI INTSERV

I principali punti negativi di IntServ è che ogni router deve mantenere uno stato per ciascun flusso e quindi ha un forte impatto sull'architettura esistente, ha una scarsa scalabilità e la segnalazione è pesante.

DIFFERENTIATED SERVICES

DiffServ rappresenta una soluzione più semplice, scalabile e di basso costo rispetto a *IntServ* (*nelle reti IP*) poiché si rinuncia al controllo stretto flusso per flusso nei router (*si usa una tecnica coarse - grained*). All'interno di ogni router viene trattata ogni **classe di servizio** collettivamente (*la QoS è garantita in termini relativi*). Può essere complementare a *IntServ* (*la riservazione delle risorse può essere effettuata per singoli flussi tramite RSVP*).

Si parla di *micro flussi* senza controllo singolare; eventuale traffico eccedente sottrae QoS a tutti i flussi appartenenti alla medesima classe di servizio e quindi è necessario, ancor più che in *IntServ*, fare regolazione del traffico.

Questo controllo viene fatto esclusivamente **ai bordi della rete**, mentre i router interni effettuano esclusivamente **forwarding differenziato** su poche classi di traffico diverse. Questo fa sì che le modifiche architetturali non siano molto significative, servono solo delle **code diverse** e degli **algoritmi di scheduling**, ma non c'è bisogno di mantenere alcun stato per alcun flusso.

La differenziazione delle classi avviene tramite il campo **Type of Service (o Differentiated Service)** dell'header IP. Degli 8 bit del capo, 6 sono usati per specificare il **Differentiated Service Code Point** che sarà interpretato dai router interni per selezionare il tipo di servizio da applicare e i vari **Per - Hop Behaviour**.

I router di bordo verificano la coerenza con gli SLA e i TCA e classificano il pacchetto definendone il DSCP, oltre a regolare il traffico.

I router interni gestiscono aggregati di traffico e trattano i macro - flussi secondo i PHB selezionati e secondo le **condizioni locali** (*ad es. esistenza di congestione*).

PHB

I più importanti Per - Hop Behaviour sono

1. **Expedited Forwarding (EF)** per applicazioni richiedenti basso ritardo e bassa latenza.
Emula una linea dedicata (*circuito*) ed è adatto a servizi real time. I ritardi e la percentuale di pacchetti persi devono essere molto bassi, quindi il traffico in eccesso rispetto al TCA non viene immesso in rete.
2. **Assured Forwarding (AF)** per applicazioni richiedenti affidabilità di consegna.
Fornisce 4 livelli di priorità in termini di allocazione di banda e di spazio nelle code, poiché si propone di evitare situazioni di congestione mediante uno scarto preventivo e differenziato dei pacchetti: ci sono 3 livelli di probabilità di scarto per ognuna delle 4 classi di qualità definite.
- Generalmente si usa un **Random Early Discarding (RED)** che scarta i pacchetti in base alle soglie dei tre livelli di scarto e si contrappone al Tail Drop che scarta i pacchetti quando la coda è piena ed è quindi incontrollato.
3. **Best Effort (BE)** in cui non si hanno garanzie.

VOICE OVER IP (VOIP)

Voice over IP è una modalità di gestione delle chiamate telefoniche che sfrutta la rete Internet invece della tradizionale rete telefonica **Public Switch Telephone Network (PSTN)** che, a differenza di IP, è una rete a circuito. In VoIP la voce viene trasmessa interamente in **formato digitale**, mentre nella rete PSTN sull'**ultimo miglio** avviene tutt'oggi una trasmissione della voce analogica tramite la rete POTS, mentre il resto è digitale.

Esistono due tipi di operatori VoIP:

1. Operatori **Over - The - Top VoIP (OTT)** che sfruttano la rete a pacchetto per offrire servizi voce (es. *Skype*). Non hanno voce in capitolo per quanto riguarda la gestione della rete.
2. **Operatori di rete** che adottano VoIP che quindi decidono come è gestita la rete.

Con VoIP si possono instaurare diverse tipologie di chiamate telefoniche e deve essere garantito l'**interlavoro**. Possiamo avere diversi scenari:

1. **VoIP - VoIP**

La comunicazione è tra due terminali intrinsecamente VoIP e i pacchetti voce sono instradati completamente nella rete Internet bypassando la rete PSTN. Si possono anche adattare dei terminali PSTN per utilizzare VoIP tramite un dispositivo che traduce il segnale di questi terminali in pacchetti.

2. **VoIP - PTSN - VoIP**

A volte il traffico voce potrebbe comunque essere instradato per la rete PTSN; è anche possibile telefonare da un dispositivo VoIP ad un dispositivo PSTN. In questi casi i pacchetti voce passeranno per un **punto di interlavoro**.

3. **VoIP - Rete Radiomobile - VoIP**

Discorso analogo al caso PTSN.

PERCHÉ VOIP?

L'integrazione di telefonia e IP rappresenta l'opportunità di realizzare un sistema globale di comunicazione rendendo IP una rete universale, richiedendo meno manutenzione. L'integrazione porta dei vantaggi sia per l'**operatore** che per l'**utente**.

La rete PTSN offre un insieme limitato di servizi, mentre IP permette di avere ad esempio servizi di schedulazione di **call forwarding**. Inoltre, con IP c'è una maggiore portabilità dell'identità su dispositivi diversi. Si può inoltre migliorare notevolmente la **multimedialità** della comunicazione e si può regolare dinamicamente la qualità delle chiamate.

CODIFICA DELLA VOCE

La codifica fornisce una **rappresentazione digitale** del segnale audio che può essere più o meno fedele. In generale, la voce umana ha delle componenti in frequenza che arrivano fino ai 20 kHz, ma la maggior parte dell'energia è concentrata nei primi 4 kHz e per questo nelle applicazioni di telefonia la voce viene campionata a 4 kHz, perdendo dell'informazione ma permettendo un minor sforzo di codifica e una rappresentazione meno pesante.

In generale si hanno tre tipi di codifica:

1. **Waveform Codec (PCM, DPCM, ADPCM)**

Codificano il segnale senza perdite (*al netto dell'errore di quantizzazione*) sulla base della forma d'onda di questo.

Viene effettuata in due passi:

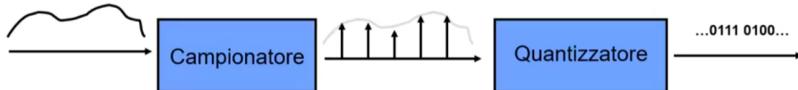
1. Campionamento

Passa da una rappresentazione del segnale tempo - continua ad una rappresentazione tempo - discreta. Dato un segnale con una banda B , secondo il teorema di Nyquist, se questo viene campionato ad una frequenza $2B$, non ci sono perdite di informazioni.

2. Quantizzazione

Rappresenta i valori dei campioni con un numero di livelli discreto e finito. Questa è un'operazione irreversibile che genera un **errore di quantizzazione**. La quantizzazione può anche essere non uniforme, andando ad addensare i livelli dove si necessita di maggiore accuratezza.

Solitamente in telefonia si usano 8 bit per campione, avendo 64 kb/s.



Una di queste codifiche è la **Pulse Code Modulation (PCM)** che usa una quantizzazione logaritmica che raffina la granularità ove le ampiezze sono più basse poiché si adatta bene alla **dinamica di loudness** dell'orecchio umano.

Esiste anche la **Differential PCM** che si basa sull'osservazione che i campioni vocali vicini temporalmente presentano della correlazione. Questa codifica usa dei **metodi di predizione** per valutare il campione successivo noti i precedenti e codifica solo la **differenza** tra il valore reale e quello predetto. Esiste anche la versione **Adattiva** che dimensiona lo step di quantizzazione sulla base dell'andamento del segnale.

2. Source Codec (*Vocoders*)

Questi codec usano dei modelli che permettono di riprodurre la voce umana sulla base delle sue caratteristiche intrinseche rimuovendone la ridondanza. Hanno un'altissima efficienza ma sono molto complessi, introducono dei ritardi elevati e sono sensibili ai rumori di fondo.

La voce è composta da due tipi di suoni: **voiced** che sono tipici nelle vocali e sono caratterizzati da un andamento periodico secondo una frequenza di **pitch** e **unvoiced** che sono tipici delle consonanti, di ampiezza ridotta e di alta frequenza.

La produzione vocale viene fatta in tre fasi:

1. **Produzione** del fiato (*aria espirata dai polmoni*)
2. **Generazione del suono** (*vibrazione delle corde vocali*)
3. **Modulazione del suono** (*risonanze del tratto vocale e degli articolatori*)

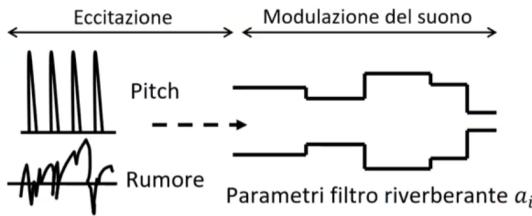
Questo processo può essere modellato per mezzo di un **modello a fonemi** (*unità base del suono*) e riprodotto tramite un **filtro riverberante a parametri discreti**.

Il vocoder di fatto effettua due fasi:

1. **Eccitazione** dove genera un treno di impulsi con un determinato pitch o **rumore bianco**.
2. **Modulazione** tramite il filtro riverberante.

Ad intervalli regolari vengono stimati e trasmessi i parametri per un certo fonema come i coefficienti del filtro lineare, se si tratta di segnale voiced o unvoiced, il pitch e il gain. Questi parametri sono scelti in modo da minimizzare l'errore tra il segnale d'ingresso e quello d'uscita.

In decodifica, il vocoder prende i parametri settati e li utilizza per sintetizzare il segnale.



3. Hybrid Codec

Sono molto simili ai vocoder lineari ma ne ottimizzano alcuni aspetti.

Si usa un **Multi Pulse Excited Linear Prediction (MPELP)** che, invece di avere due eccitazioni, permette di avere un unico tipo di eccitazione con N impulsi che però possono avere ampiezze diverse e possono non essere equispaziati (es. codificatore *GSM*).

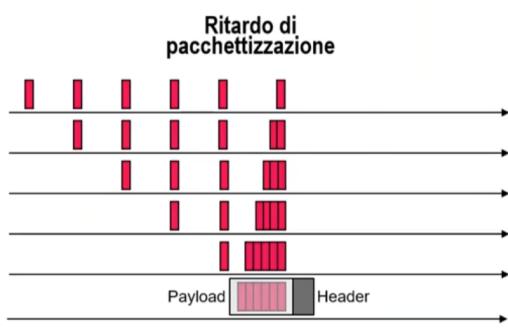
Esiste anche la **Code Excited Linear Prediction (CELP)** dove la sequenza di eccitazione viene scelta da un insieme di sequenze (*codebook*) minimizzando così l'errore. Causa però più ritardi per la ricerca della sequenza migliore nel codebook.

Per la valutazione di una certa codifica vocale si usa il **Mean Opinion Score (MOS)** che è una misura **soggettiva** e quindi misura la *Quality of Experience* ed è basata sull'opinione di un gran numero di ascoltatori. Generalmente i waveform codec hanno qualità maggiori a discapito di un maggior consumo di dati.

DEGRADO DELLA VOCE

In VoIP, oltre che dalle caratteristiche intrinseche della codifica, la qualità è influenzata anche dalla trasmissione in rete e in particolare dai **ritardi** introdotti e dalla perdita totale o parziale di pacchetti (*anche se questo è un problema minore*).

I ritardi in IP sono causati principalmente dalla pacchettizzazione, dalla propagazione dei pacchetti (*con relativo accodamento*) e dal ritardo di playout.



Il ritardo di pacchettizzazione è causato dall'accumulo di più segmenti vocali in un unico pacchetto per la trasmissione sulla rete a pacchetti. Questo ritardo dipende dalla lunghezza dei segmenti e dal numero di segmenti. Se i segmenti sono di dimensione maggiore, ne pacchettizzeremo di meno generando meno ritardo.

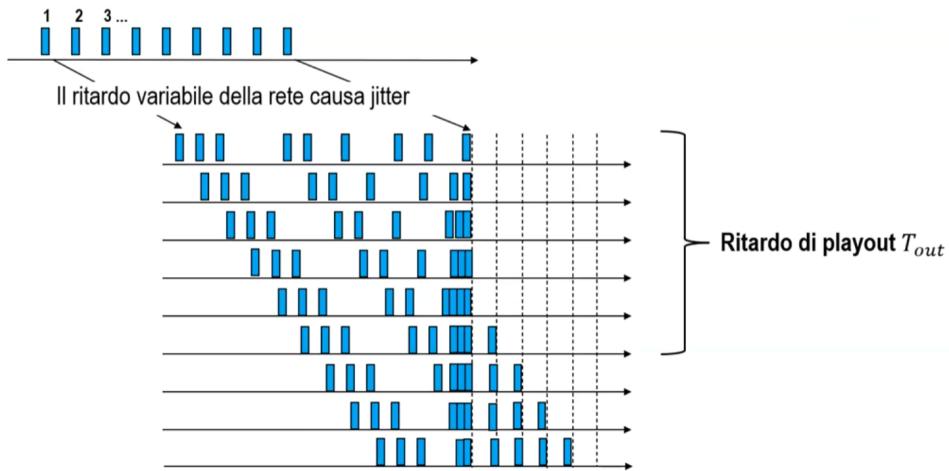
Il ritardo di trasmissione dipende invece dalla velocità del link e dalla dimensione dei dati che invio e si verifica ad ogni invio. Non dipende in alcun modo dalla distanza.

Il ritardo di propagazione invece da quanto rapidamente il mezzo trasmittivo propaga il segnale e dipende quindi dalla distanza.

Esiste anche un **ritardo di elaborazione** dei pacchetti da parte di switch e router ma solitamente è trascurabile se gli switch/router sono ben dimensionati. Inoltre c'è anche un **ritardo di accodamento** dei pacchetti nei buffer che non è presente nelle reti a circuito come PTSN.

Il ritardo di playout è un ritardo introdotto volutamente per ridurre il **jitter**, ovvero la variazione del ritardo di ricezione dei pacchetti emessi.

Valori elevati di jitter danno problemi nella riproduzione audio. Per fare questo i pacchetti vengono accodati in un buffer di playout.



MIGLIORARE L'EFFICIENZA

Nelle conversazioni duplex il canale è usato in media per il 50% del tempo, mentre per il resto ci sono *silenzii*. La **soppressione** dei silenzi viene usata nelle reti a pacchetto proprio per ridurre la banda media occupata. Tuttavia, i silenzi devono essere soppressi senza dare l'impressione di caduta di linea e quindi si è introdotto il rumore di fondo (*lato ricevitore*).

SEGNALAZIONE VOIP

I protocolli di segnalazione VoIP hanno il compito di **controllare, aprire e chiudere** le chiamate (*flussi dati*) e non sono direttamente coinvolti nella trasmissione delle chiamate perché fanno parte del piano controllo. Esiste la segnalazione **in banda** dove la segnalazione usa lo stesso canale delle chiamate e la segnalazione **fuori banda** che usa un canale dedicato.

Nella rete PTSN la segnalazione (SS7) riguarda esclusivamente le chiamate e coinvolge chiamante, rete che instrada e prenota i circuiti e chiamato. Si fanno poi dei controlli d'accesso e della predisposizione della documentazione d'addebito.

Sulla rete IP la segnalazione può essere ridotta al minimo poiché l'instradamento dei flussi dati è effettuato normalmente dal protocollo IP e posso essere usati meccanismi esistenti per traslare un indirizzo IP nel nome del chiamante/chiamato (es. DNS).

Potrebbe bastare l'aggiunta di un protocollo di allerta per l'utente chiamato e un protocollo di negoziazione dei parametri di sessione (*codec, numeri, media supportati*). In realtà però i protocolli di segnalazione specifici rimangono necessari per gestire al meglio alcune funzionalità come controllo degli accessi, controllo delle risorse, tariffazione, ecc.

Esistono due principali architetture di segnalazione VoIP e sono **H.323** e **Session Initiation Protocol (SIP)** che è la più diffusa.

SESSION INITIATION PROTOCOL

Questo protocollo è nato nel 2002 e può essere utilizzato per instaurare chiamate audio o audio/video. Consente le chiamate tra utenti per mezzo di identificativi che prendono il nome di **Universal Resource Identifier (URI)**. Somiglia molto all'e-mail address dal punto di vista semantico e sintattico.

Un esempio di identificativo è `user@domain.tld` dove user può anche essere numerico per

rimanere compatibili con lo standard di numerazione usato nella rete PTSN. L'identificativo è dell'utente e non del terminale e questo può accedere da terminali diversi e per questo sono richieste funzioni di **registration** e **user location**.

Il SIP è un protocollo applicativo (*livello 7*) di tipo client - server. Ogni nodo della rete coinvolto nella segnalazioni SIP implementa sia client che server.

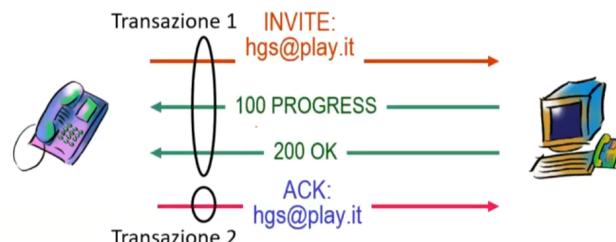
Le **richieste** dei client si chiamano **metodi**. Queste possono essere trasmesse perché non si assume l'affidabilità nella consegna.

INVITE	Inizia la chiamata invitando un utente
ACK	Conferma la avvenuta connessione
BYE	Termina (o trasferisce) la chiamata
CANCEL	Cancella una richiesta precedente
OPTIONS	Chiede informazioni
REGISTER	Registra presso il location service

Le **risposte** hanno un **codice** che descrive l'esito dell'elaborazione della richiesta da parte del server che seguono il modello HTTP (*100 info, 200 OK, ecc.*).

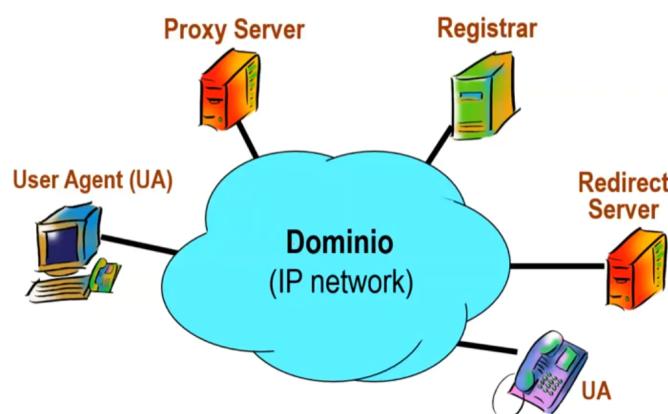
In SIP richieste e risposte sono raggruppate in differenti **transazioni SIP** e hanno l'obiettivo di instaurare una **sessione**, ovvero uno scambio interattivo di media. Una transazione è composta da una richiesta, eventuali risposte informative (*1xx*) e una risposta finale.

Per i messaggi di INVITE è necessario un ACK finale che confermi l'apertura della sessione che di fatto è una transazione a sé stante senza risposta.



SIP, essendo un protocollo applicativo, è **character - oriented**, ovvero è testuale e i messaggi SIP sono *human readable*. Usa dei campi *parameter : value*. Si ha una *Start Line*, un *Header*, una linea vuota e il corpo del messaggio. Nelle risposte si possono avere delle variazioni nell'header.

SIP suddivide la rete in **domini** e ogni dominio include vari elementi di rete.



- **USER AGENT (UA)**: Rappresenta il mittente (*client*) o il destinatario (*server*) di una sessione. Ad esempio un internet phone o un software per teleconferenza.

- **REGISTRAR:** Questo è un nodo che, all'interno di un dominio, associa lo URI dell'utente all'indirizzo dello UA su cui l'utente può essere rintracciato. Per essere raggiungibile verso uno specifico UA l'utente deve registrarsi presso il registrar inviando una richiesta di REGISTER contenente URI e IP address dell'UA.

L'utente può ottenere un indirizzo IP in tre modi:

1. Configurazione statica
2. Utilizzo di servizi di traslazione degli indirizzi (*DNS*)
3. Inviando a richiesta di REGISTER ad un indirizzo di multicast

- **PROXY SERVER:** Il proxy è come un *router* di livello applicativo che instrada le richieste e le risposte tra differenti domini. SIP garantisce che le risposte seguano a ritroso la strada delle richieste grazie al campo **Via** che indica i vari proxy attraversati. A ritroso il rispettivo campo viene rimosso.

Si può fare anche con **Record-Route** e **Route** che però non vengono rimossi in modo che la sorgente conosca la route conosciuta.

Il Proxy Server può inoltrare le richieste anche a più destinazioni (*automatic call distribution*) e può avvenire sia in serie che in parallelo e viene mantenuta la prima risposta positiva e, se necessario, si invia una CANCEL agli altri UA. Si parla di **SIP Forking**.

- **REDIRECT SERVER:** Riceve le richieste e risponde con l'indicazione della localizzazione di un altro UA. Solitamente sono implementati direttamente nell'UA. Permette di non coinvolgere il Registrar quando ad esempio l'utente si è spostato solo temporaneamente.

SESSION DESCRIPTION PROTOCOL

SIP usa **SDP** per la descrizione delle sessioni e i messaggi SDP sono trasportati in maniera trasparente come corpo dei messaggi SIP. Solitamente si ha un messaggio SDP nel messaggio INVITE. I messaggi SIP contengono informazioni sul nome e scopo della sessione, sulla durata, i media utilizzati, gli indirizzi sempre del tipo *parameter = value*.

Esistono anche i messaggi **PRACK** che serve a dire al server che il client sta lavorando mentre questo aspetta un ACK e **UPDATE** per modificare solo i media. Per modificare la sessione è necessario un nuovo INVITE.

Si possono fare trasferimenti di chiamata col metodo **REFER** e si possono costruire servizi di notifica di eventi con il metodo **NOTIFY** (*ad esempio si notifica l'utente della vecchia sessione per ogni evento nella nuova sessione*).

SIP è utilizzato nelle reti radiomobili di generazione recente (*4G e 5G*) dove si usa una versione ottimizzata di VoIP, VoLTE in modo che consumino meno banda.

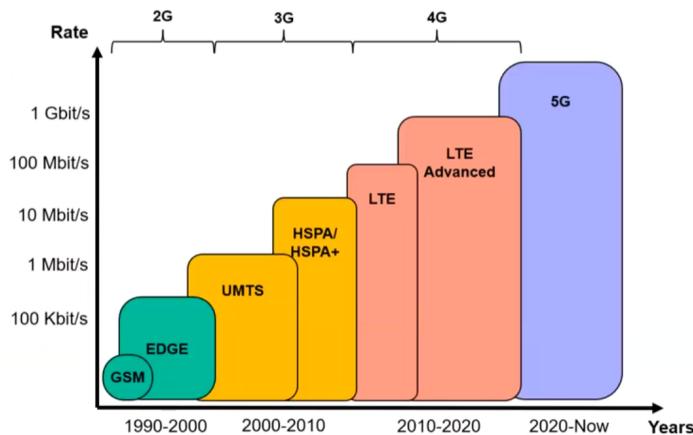
RETI RADIOMOBILI

Nel 1973 venne fatta la prima chiamata su una rete radiomobile da *Martin Cooper* con un telefono che pesava circa 1 KG (*il mattone*). Nel 1979 l'operatore giapponese NTT crea la sua prima rete: cinque anni più tardi diventa la prima **rete 1G** a coprire tutto il paese.

Nel 1981 nasce in Scandinavia una rete radiomobile che permette anche di fare roaming. Nel 1990 nasce **GSM** in Europa, una rete **2G** standardizzata che poi diventa un servizio commerciale. Nel 1999 nasce **WAP**, uno standard per fruire di contenuti Web testualmente su telefoni. Nel 2001 nasce poi **3G** e nel 2003 nasce il primo PDA con un sistema operativo e applicazioni che però non aveva una rete sua, si connetteva solo tramite WiFi. Nel 2007 si ha la vera rivoluzione con il primo

iPhone che introduce di fatto il concetto di dispositivo connesso con applicazioni.

Nel 2009 nasce il **4G** e nel 2018 il **5G**.



INFRASTRUTTURA

In generale in una rete radiomobile è una rete che permette a degli *user terminals (User Equipment o UE)* di connettersi a dei servizi di telefonia ed Internet. Sono dei tipi specifici di reti wireless che seguono una **architettura ben definita** e che permettono una comunicazione **seamless** durante la mobilità dell'utente.

Una rete radiomobile ha due componenti fondamentali:

1. **Radio Access Network (RAN)**: gestisce la comunicazione radio con gli *UE*.

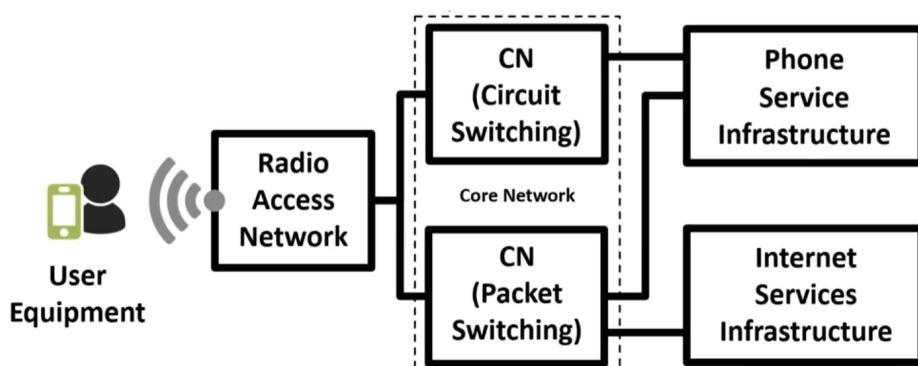
Le sue componenti principali sono le **Base Stations (BS)** che si connettono agli *UE* tramite un **interfaccia radio** e si connettono alla **Core Network** tramite una **backhaul network**. L'area dove il servizio è offerto è diviso in **celle**, ognuna coperta da una Base Station (*hanno forma più o meno esagonale in modo da coprire perfettamente il piano*).

Ci sono delle procedure per la gestione della mobilità come **Cell Selection, Location Update, Handover e Paging**.

La procedura dipende dallo stato del *UE* (*idle mobility o active mobility*).

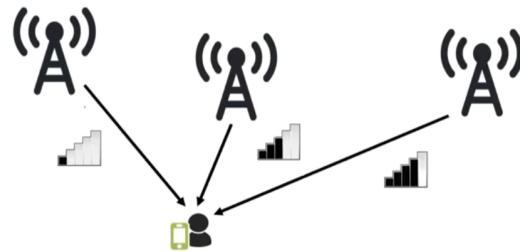
2. **Core Network (CN)**: interconnette la *RAN* alle infrastrutture esterne garantendo connettività e gestione della mobilità.

Possono essere **Circuit - Switching** (*usate per fornire servizi di fonia nelle reti 2G e 3G*) o **Packet - Switching** (*forniscono accesso ai servizi dati e nelle reti 4G e 5G anche ai servizi di fonia*).



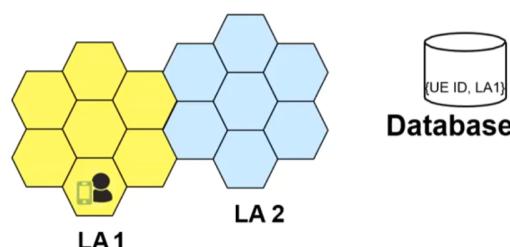
Vediamo in dettaglio le procedure per la gestione della mobilità:

1. Cell Selection: quando sono in stato di **idle mobility**, gli *UE* scelgono autonomamente la base station più conveniente ascoltando un segnale a massima potenza che ogni *BS* invia (**beacon**). Solitamente si sceglie la *BS* il cui beacon presenta la maggior potenza.



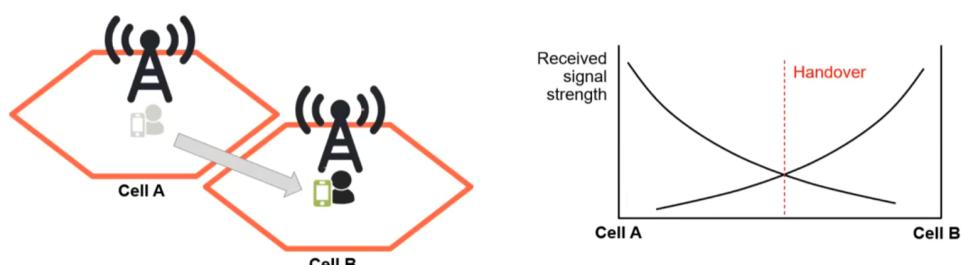
2. Location Update: in **idle mobility**, la posizione degli *UE* è tracciata con una granularità di **Location Area (LA)** pari a un set di celle contigue identificate da un codice univoco. La *Location Area* di ogni *UE* è salvata in un database.

Quando un *UE* si sposta da una *LA* ad un'altra viene avviata la procedura di **Location Update** dall'utente e la *LA* viene aggiornata nel database in modo da instradare le chiamate o le sessioni di dati correttamente al *UE*.

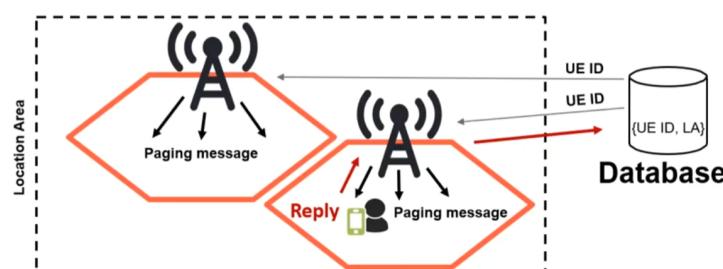


3. Handover: quando una *UE* è in **active mobility** la sua location è tracciata con una **granularità di cella**. La procedura di mobility in questo caso si chiama *handover* ed è **gestita dalla rete** (*ma assistita dal terminale*) che adatta il routing e impone al *UE* di cambiare cella quando una nuova connessione è stata instaurata (**make-before-break**).

Per evitare un continuo switching tra le celle si attende del tempo in cui il segnale della nuova cella è più potente di quella vecchia prima di fare effettivamente l'handover.



4. Paging: quando una sessione di fonia/dati deve essere indirizzata a un *UE* che è in stato **idle**, si ricava dal database la *LA* e il *UE ID* e si inizia la procedura di **paging** dove tutte le *BS* della *LA* inviano un **paging message** in broadcast che include il *UE ID*. Il *UE* risponde al *messaggio di paging* e in questo modo la sessione di fonia/dati può essere instradata alla giusta *BS* cambiando lo stato del *UE* in **active**.



DOMANDA: qual è la dimensione più conveniente per una Location Area?

Dipende: la dimensione delle *LA* include il tradeoff tra due bisogni conflittuali:

- *LA* più grandi implicano un maggiore traffico dovuto al **paging**, ma necessita di meno **location updates**.
- *LA* più piccole implicano più traffico dovuto ai **location updates**, ma necessita di meno **paging**.

La dimensione ottimale dipende dal traffico in ingresso e dalla mobilità degli utenti.

RADIO PLANNING

Il *Radio Planing* è l'operazione che decide le location e le configurazioni ottimali delle *BS* per coprire al meglio un'area geografica ed evitare il più possibile le interferenze tra celle differenti. Si hanno due fasi:

1. **Coverage Planning:** seleziona dove installare le *BS* e decide le loro configurazioni.

Si considerano quattro aspetti fondamentali:

1. Propagazione predetta del segnale

Questo è importante perché permette di stimare l'area vera coperta dalla *BS* (*area coperta = area dove il segnale è maggiore rispetto a quello delle altre BS*). La potenza del segnale dipende dalla **potenza emessa** e dal **path loss** che dipende dalle frequenze, dall'urbanizzazione e dalla morfologia del terreno.

2. Stima del traffico

Dipende chiaramente dalla popolazione, dall'urbanizzazione, dal mercato del servizio considerato, ecc.

3. Posizionamento della *BS*

Un set di posizioni candidate viene definito in base ad **aspetti tecnici** come la stima del traffico, la morfologia, ecc. e in base ad **aspetti non-tecnici** come l'inquinamento elettromagnetico, accordi con proprietari di palazzi, leggi locali, ecc.

4. Configurazione antenna

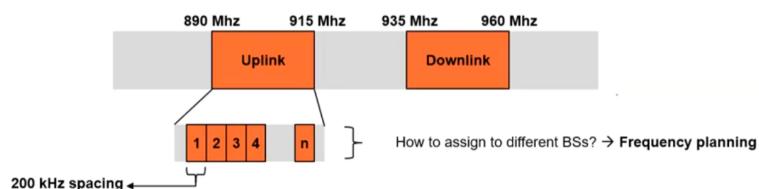
Le antenne vengono configurate con diagrammi di radiazione, tilt rispetto al piano orizzontale, massima potenza emessa, altezza, capacità, ecc.

Diverse configurazioni portano a diverse coperture.

Si usano poi dei **test point** e dei **modelli di ottimizzazione matematica** per minimizzare i costi garantendo comunque una qualità minima per tutti i test point. Le considerazioni fatte in questa fase non tengono conto delle interferenze tra *BS* e assumono che tutte le risorse sono assegnate a tutte le celle.

2. **Frequency Planning:** decide come assegnare le risorse radio (*frequenze*) alle varie *BS*.

In *GSM* si hanno due bande, una per l'**uplink** e una per il **downlink** suddivise in **sub - carrier** che possono essere associate a diverse *BS*.



Tuttavia, essendo i sub - carrier in numero limitato, è necessario riutilizzare le stesse frequenze per diverse celle almeno in parte.

L'obiettivo è minimizzare l'interferenza.

Per farlo, si assegnano frequenze diverse a celle vicine. Un gruppo di celle adiacenti che usano diverse frequenze si chiama **cluster** e questi *cluster* si ripetono regolarmente in modo che ogni frequenza possa essere riutilizzata una volta per frequenza. Chiaramente un *cluster* maggiore significa meno interferenze ma meno risorse per cella.

I *cluster* hanno dei possibili valori predefiniti k e il l'efficienza di riuso delle frequenze è calcolata come $1/k$.

I valori di k ammissibili sono tali per cui una cella ha altre sei celle che utilizzano la sua stessa frequenza alla stessa distanza da essa.

Nelle nuove tecnologie si usano tecniche di modulazione come *OFDM* e *CDM* che permettono di avere *cluster size* pari a 1.

L'utilizzo di **antenne settoriali** può ridurre l'interferenza e aumentare il riuso delle frequenze andando a mettere un'antenna al centro di una cella o, più comune, ai bordi di tre celle. Tuttavia, le antenne settoriali producono anche delle interferenze nelle celle adiacenti per via dei **side lobes**.

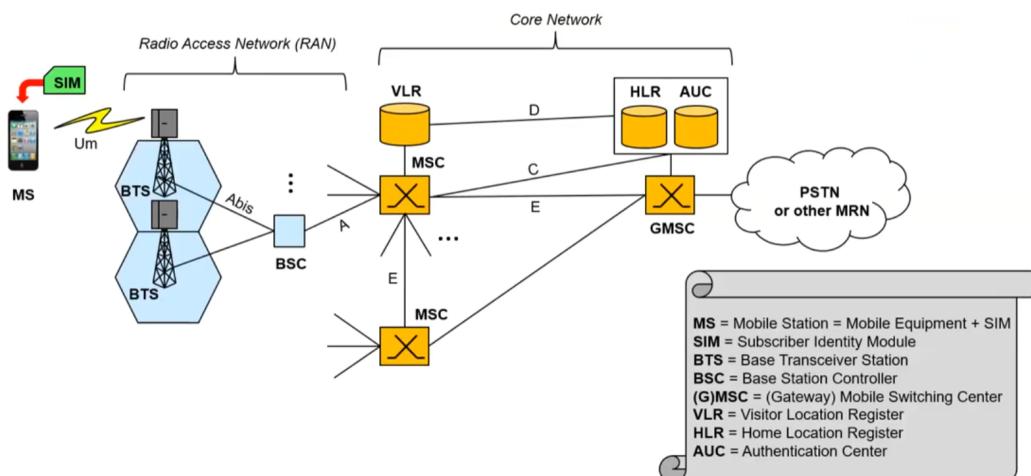
Inoltre le frequenze adiacenti spesso hanno una parte di spettro in overlap tra loro e quindi spesso è meglio non assegnare canali adiacenti alla stessa cella.

A volte è possibile anche avere dimensioni diverse per le celle in base al traffico stimato andando così a organizzare meglio le risorse e si può fare anche **cell splitting** nel caso il traffico aumenti in una zona in cui c'è una cella grande per aumentare la capacità disponibile.

Chiaramente però con celle più piccole si rischia di avere un numero elevato di *handover*. Questo si risolve con un sistema gerarchico andando a sovrapporre una cella maggiore alle celle piccole che sarà assegnata agli utenti con una grande mobilità, lasciando le celle piccole agli utenti con bassa mobilità.

2G - GSM

Global System for Mobile Communications è la seconda generazione di rete cellulare e la prima digitale. L'architettura di GSM è **circuit - based** ed è quindi stata pensata per la fonia più che per i dati.



In GSM abbiamo le **Mobile Stations (UE)** che includono un *Terminal Equipment* e un *Subscriber Identity Module* card.

Le **BS** sono **Base Transceiver Stations (BTS)** che gestiscono la connettività a livello fisico con gli *UE* e eseguono comandi di riallocazione delle risorse ricevuti dal **Base Station Controller** che gestisce le risorse radio (*di solito un BSC gestisce una intera Location Area*).

Abbiamo poi il **Mobile Switching Center (MSC)** che include funzioni di controllo e di segnalazioni e gestiscono la mobilità.

Abbiamo poi il **Visitor Location Register (VLR)** che è il database associato al *MSC* che contiene le informazioni degli *UE*.

Si ha un **Gateway MSC** che si interconnette alla rete PTSN o ad altre reti radio e quindi svolge attività di interlavoro.

L'**Home Location Register** è un database centrale che mantiene dati permanenti sugli utenti e dati dinamici relativi al VLR sul quale sono presenti i dati dell'utente.

L'**Authentication Center (AUC)** autentica le SIM che cercano di connettersi alla rete core GSM.

CANALI

In GSM si usano FDM e TDM insieme per la divisione delle risorse radio. Lo spettro delle frequenze è diviso in diverse porzioni (**sub - carrier**) e su ogni porzione è applicata la TDM creando dei **timeslot** usati per i dati di diversi **canali**.

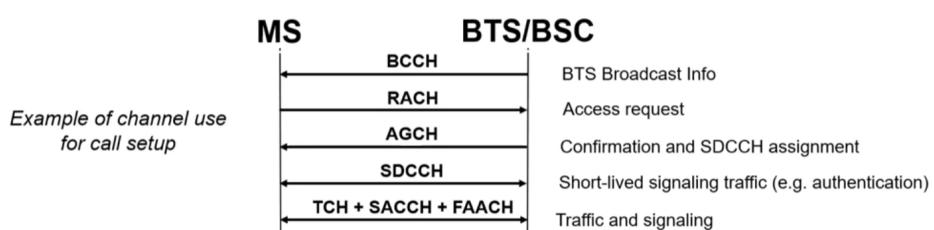
Abbiamo dei **canali di traffico (TCH)** usati per trasportare dati voce e diversi **canali di controllo** come **Broadcast Control Channel (BCCH)** che trasporta informazioni generali sulle BTS come l'ID e la *Location Area* e funziona anche da **beacon channel** e quindi trasmette sempre a massima potenza. Un altro canale di controllo è il **Paging Channel (PCH)** che è usato dalle BTS per le procedure di paging quando un MS riceve una chiamata. Questo canale viene usato anche per gli SMS.

Si ha anche un canale **Random Access (RACH)** in uplink che è contention - based (*slotted Aloha*) e viene usato dalle MS per accedere alla rete ogni volta che deve avviare una nuova procedura e richiedere nuove risorse.

Si ha il canale **Access Grant** che è usato per rispondere alle richieste fatte sul canale *RACH* e include informazioni su come usare il canale *SDCCH*.

Il canale **Stand - Alone Dedicated Channel (SDCCH)** è usato per le procedure short - lived come autenticazione, location update, ecc.

Ci sono infine il **Slow Associated Control Channel (SACCH)** usato per scambiare misure durante la connessione tra *MS* e *BTS* (*es. potenza segnale*) e **Fast Associated Control Channel (FAACH)** che trasporta i segnali generati durante l'handover e la call setup.



IDENTIFICATORI

GSM presenta cinque tipi di identificatori:

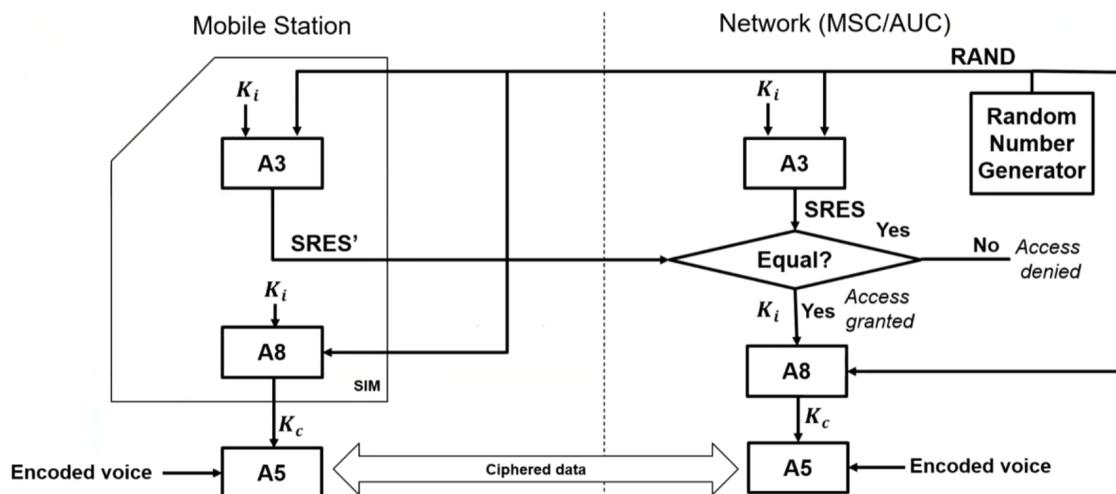
1. **Mobile Station ISDN Number**: il numero di telefono permanente e permette di raggiungere la rete dell'operatore.
2. **Mobile Station Roaming Number**: numero temporaneo associato per inoltrare una chiamata roaming al MSC che la gestisce (*che non potrebbe trovare l'utente che appartiene ad una rete diversa*).
3. **International Mobile Station Identifier**: identifica permanentemente l'MS ed è scritto nella SIM e salvato nel HLR. Questo viene inviato solo se non è disponibile un *TMSI* già associato.
4. **Temporary Mobile Station Identifier**: identifier dinamico allocato dal VLR, salvato nel VLR e nella SIM. Creato per ragioni di sicurezza.
Ad ogni location update il *TMSI* cambia.
5. **Location Area Identity**: indica la *LA* ed è salvato nella SIM e nel VLR.

AUTENTICAZIONE

Le operazioni di sicurezza (**autenticazione e cifratura**) sono gestite dalla SIM Card e dal MSC / AUC.

Si usano dei parametri e degli algoritmi:

- **K_i**: chiave univoca di autenticazione da 128 bit immagazzinata nella SIM e nel AUC.
- **RAND**: numero random da 128 bit generato dal AUC e inviato al *UE* tramite il *MSC*.
- **A3**: algoritmo per l'autenticazione immagazzinato sia nella SIM che nel AUC.
- **A5**: algoritmo di cifratura di flusso immagazzinato nel *UE* e nella *BS*.
- **A8**: algoritmo di generazione di chiavi **K_c** presente nella SIM e nel AUC.



Nella procedura di autenticazione si genera un numero random nella rete (*challenge*) che sarà inviato alla SIM che lo cifrerà con la chiave *K_i* (*che è un segreto condiviso tra SIM e rete*) e risponderà alla challenge. La rete effettuerà la stessa operazione e controllerà che il risultato sia lo stesso.

Se è lo stesso, si procede con la generazione della chiave per la cifratura della comunicazione utilizzando come parametri *K_i* e il numero random calcolato prima e poi la comunicazione può avvenire in maniera cifrata dalla chiave *K_c*.

Questa operazione viene rieffettuata periodicamente in modo che *K_c* sia cambiata.

Ovviamente si autentica la SIM, non l'utente. Inoltre è presente una falla in questo sistema: l'utente si autentica nei confronti della rete, ma la rete non si autentica nei confronti dell'utente e quindi sono possibili attacchi del tipo *Man - in - the - Middle* tramite degli **IMSI - Catcher**.

ACCENSIONE UE

Quando un *UE* viene acceso si esegue la seguente procedura:

1. **Cell Selection:** il *UE* seleziona la *BS* a cui collegarsi (*col segnale più potente*).
2. **Registration:** il *UE* avverte il *MSC* che è attivo e si hanno due casistiche:

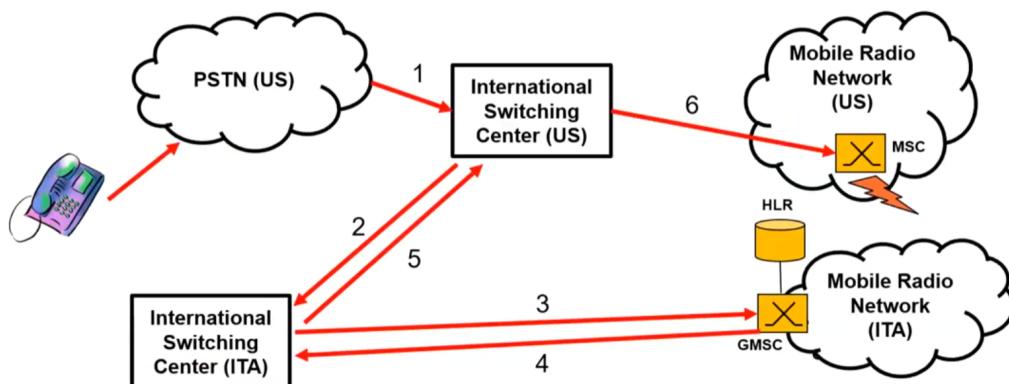
1. **IMSI Attach:** l'id della *LA* ricevuto dalla rete è lo stesso immagazzinato nella SIM.
L'*IMSI* viene impostato come attivo nel *VLR*.
2. **Location Update:** nessun *LA* ID immagazzinato o *LA* ID diversi rete / SIM
Si procede con una procedura di *Location Update*.

Quando si effettua la *Location Update* si hanno due casi:

1. Le due *LA* sono servite dallo stesso *MSC*
In questo caso non è necessario allocare nessun nuovo *TMSI*, ma viene semplicemente aggiornato il *LA* ID nel *VLR*.
2. Le due *LA* sono servite da diversi *MSC*
In questo caso è necessario cambiare *TMSI* e allocarne uno nuovo.
In questo caso l'utente chiede al nuovo *MSC* di cambiare *LA* ID e questo chiederà al proprio *VLR* di chiedere a quello vecchio il **IMSI** associato al **TMSI** inviato dall'utente. Viene poi fatta l'autenticazione della SIM e infine aggiornata la *LA* e inviato il nuovo *TMSI* all'utente.
Infine il *VLR* nuovo informa l'*HLR* che l'utente è gestito nel nuovo *VLR* e questo farà cancellare al *VLR* vecchio le informazioni sull'utente.

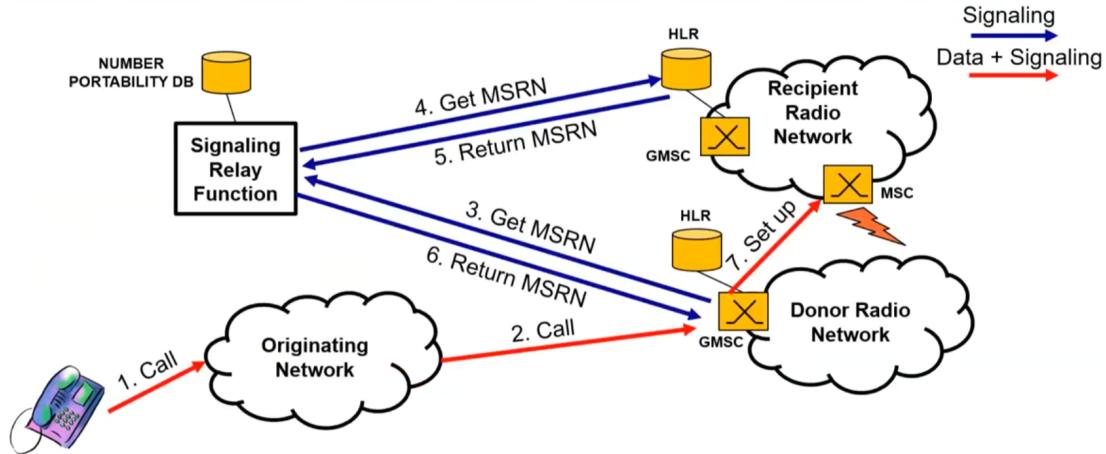
ALTRE PROCEDURE

Per la *Call Setup* si usa una procedura di segnalazione molto simile a quella adottata in *PSTN*. Tuttavia *GSM* presenta una certa inefficienza per quanto riguarda le chiamate in *roaming*: se un utente dagli stati uniti chiama un numero italiano che però si trova in roaming negli stati uniti, tutta la comunicazione passa comunque per il *GMSC* italiano facendo avanti e indietro.



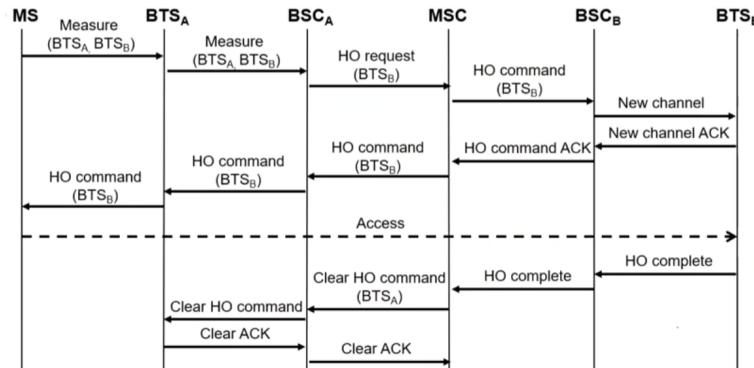
Quando viene fatta una portabilità del numero vengono inviati dei messaggi di segnalazione ulteriori ad un **number portability db** che permette la comunicazioni tra la rete donatrice e quella recipiente del numero per poter instradare correttamente la chiamata.

In questo caso, rispetto al precedente, la voce segue un percorso più efficiente rispetto ai messaggi di segnalazione.



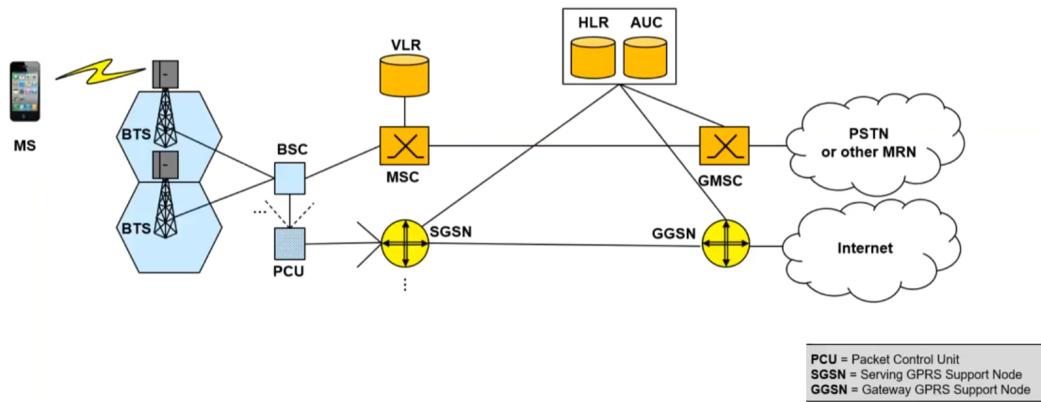
Per quanto riguarda l'handover, questo può avvenire *Intra BSC*, *Inter BSC* (ma *intra MSC*) o *Inter MSC*.

Ci focalizziamo sul handover *Inter BSC*: in questo caso la *BSC* attuale fa una richiesta a quella nuova di handover tramite il *MSC*. La nuova *BSC* allora alloca quindi un canale sulla nuova *BS* su cui eseguire l'handover e informa l'utente che deve switchare *BS*.



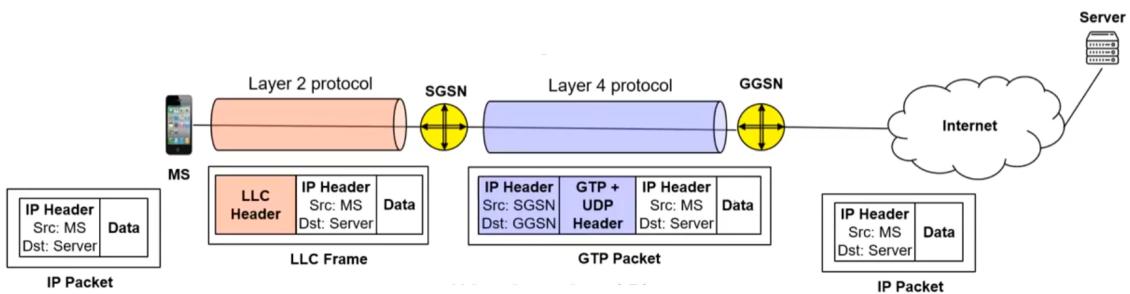
2G - GPRS

General Packet Radio Service (**GPRS**) è un servizio di commutazione a pacchetto costruita sopra GSM. Alcuni canali dedicati al traffico sono condivisi dinamicamente in uplink e downlink da diversi *UE* e sono acceduti tramite Slotted Aloha: questi canali si chiamano **Packet Data Traffic Channel (PDTCH)**. Sono anche definiti nuovi canali di controllo per supportare la rete a pacchetti. Un'evoluzione di GPRS è **Enhanced Data Rate for Global Evolution (EDGE)** che apporta delle modifiche fisiche(*nuove modulazioni*) per aumentare la velocità fino a 270 Kbps.



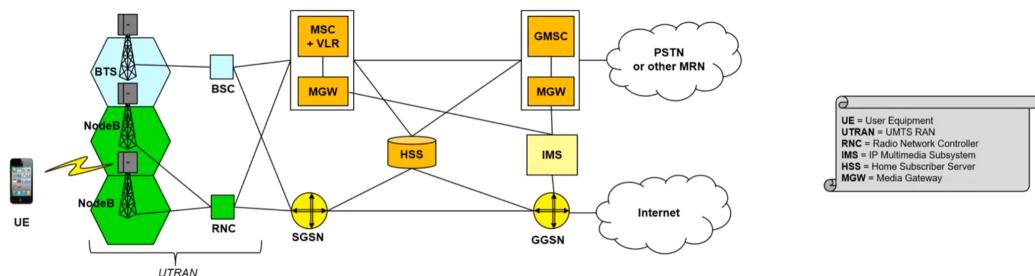
Alla rete vengono aggiunti un **Packet Control Unit (PCU)** e dei *router* con funzioni aggiuntive per gestire la mobilità.

GPRS fa un packet forwarding basato sui **tunnel**: in particolare si usano un tunnel di livello 2 (LLC) tra *UE* e *SGSN* e un tunnel di livello 4 (GTP) tra *SGSN* e *GGSN*. Questi tunnel sono usati per gestire la mobilità dell'utente e per eseguire la multiplazione dei vari flussi.



3G - UMTS

Universal Mobile Telecommunication System è la prima tecnologia 3G e apporta delle modifiche nella **Radio Access Network** introducendo delle interfacce **CDMA**, i **bearer services** e un **soft handover**. La core network è molto simile a quella di GPRS, anche se si pianifica di farla evolvere verso una soluzione *all IP* (*processo che si concluderà in LTE*).



I nodi nuovi più importanti sono l'**IP Multimedia System (IMS)** che permette di stabilire sessioni multimediali tra terminali IP usando una segnalazione SIP e il **Media Gateway (MGW)** che converte gli stream media tra diverse tecnologie (2G, 3G).

UMTS permette di avere una velocità di 2 Mbps e in HSPA e HSPA+ si arriva fino a 14.4 e 42 Mbps grazie a improvmenti nella codifica, nella modulazione e nell'adozione di antenne MIMO e della *flat network* dove i messaggi di controllo e quelli dati seguono percorsi differenti.

La *UTRAN* permette di assegnare risorse a canali con diverse caratteristiche tramite i **bearer**, ovvero delle *bit pipe* con certi attributi collegati alla *QoS* a cui sono associati dei parametri come massimo bitrate, bitrate garantito, delay, probabilità d'errore. Esistono inoltre diversi bearer services tra diversi nodi come i **Core Network Bearer Services** tra i nodi della core network (*SGSN e GGSN*), i **Radio Access Bearer Services** tra *UE* e *SGSN* e i **Radio Bearer Services** tra *UE* e *RNC*.

SOFT HANDOVER

Il soft handover permette al *UE* di collegarsi a multiple *BS* (*che viene messo in soft-handover state*) in simultanea (*fino a sei*) e comunica con tutte mantenendo il primo messaggio ricevuto correttamente. Presenta un vantaggio in quanto la qualità della comunicazione è migliore e non ci sono interruzioni di servizio ma richiede chiaramente più risorse radio.

AUTENTICAZIONE

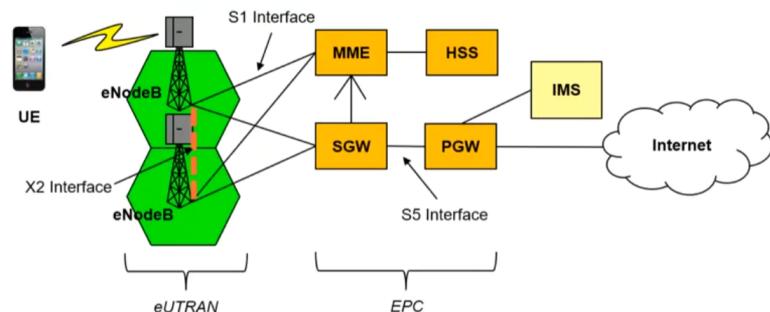
In UMTS inoltre si risolve la falla di autenticazione del 2G e si richiede che anche la rete si identifichi al utente con una doppia challenge.

4G - LTE

Long Term Evolution effettua un redesign della rete per supportare il paradigma *all IP* andando a non supportare più le reti a circuito.

Si fissa anche il concetto di **flat network** dove i dati seguono un percorso diretto diverso dal percorso dei dati di segnalazione. Inoltre si riducono i nodi dell'architettura semplificandola e tutti i nodi sono a commutazione di pacchetto.

Si hanno inoltre dei significativi miglioramenti a livello fisico con l'adozione della modulazione multi - carrier **OFDMA/OFDMA**, di modulazione e codifica adattive e di sistemi MIMO.



Tutti i vari controllori vengono quindi integrati nei nodi e le *BS* adiacenti ora dispongono di un'interfaccia di comunicazione diretta tra loro (*usata ad esempio per l'handover*). Il nodo principale che gestisce i messaggi di controllo ora è il **Mobility Management Entity (MME)**.